



Pós-Graduação em Ciência da Computação

LUIZ HAMILTON ROBERTO DA SILVA

**LOG DE EVENTOS: APLICAÇÃO DE UM MODELO
DE ANÁLISE DE LOGS PARA AUDITORIA DE
REGISTRO DE EVENTOS**



Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE
2017

Luiz Hamilton Roberto da Silva

Log de Eventos: Aplicação de um modelo de análise de logs para auditoria de registro de eventos

Este trabalho foi apresentado à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre Profissional em Ciência da Computação.

ORIENTADOR: Prof. Carlos André G. Ferraz

RECIFE
2017

Catálogo na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

S586l Silva, Luiz Hamilton Roberto da
Log de eventos: aplicação de um modelo de análise de logs para auditoria de registro de eventos / Luiz Hamilton Roberto da Silva. – 2017.
112 f.: il., fig., tab.

Orientador: Carlos André Guimarães Ferraz.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2017.
Inclui referências e apêndices.

1. Ciência da computação. 2. Auditoria de *logs*. I. Ferraz, Carlos André Guimarães (orientador). II. Título.

004

CDD (23. ed.)

UFPE- MEI 2017-238

Luiz Hamilton Roberto da Silva

Log de Eventos: Aplicação de um modelo de análise de logs para auditoria de registro de eventos

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Mestre Profissional em 14 de julho de 2017.

Aprovado em 14/ 07/ 2017.

BANCA EXAMINADORA

Prof. Kelvin Lopes Dias
Centro de Informática / UFPE

Prof. Obionor de Oliveira Nóbrega
Universidade Federal Rural de Pernambuco / UFRPE

Prof. Carlos André Guimarães Ferraz
Centro de Informática / UFPE
(Orientador)

*Dedico esta conquista e esforço à minha
família, em especial à minha Mãe
Alaídes Silva e ao meu Pai Adolfo
Ribeiro, os quais são os responsáveis
por minha chegada a este mundo.
Dedico, também, à minha Esposa Kelly
Roniele e aos meus Filhos Luiz Ricardo
e Layz Roniele, pela caminhada nesta
vida e pelo apoio, sempre,
incondicional.*

AGRADECIMENTOS

A **Deus nosso Eterno Protetor Benevolente**, o Qual nos permite estarmos aqui neste plano e nos dedicarmos a alcançar um pouco do conhecimento terreno.

A todos **meus familiares**, em especial à minha Mãe **Alaídes Roberto da Silva** e ao meu Pai **Adolfo Ribeiro da Silva**, pelo suporte no início de minha vida.

À minha Esposa **Kelly Roniele Oliveira da Silva**, que vem mudando positivamente a minha vida há 15 anos e, aos meus Filhos **Luiz Ricardo Roberto Oliveira da Silva** e **Layz Roniele Roberto Oliveira da Silva**, pelo carinho, confiança e humanidade que demonstram desde a mais tenra idade.

Agradeço aos **Professores do Centro de Informática** da Universidade Federal de Pernambuco (**Cin/ UFPE**), pelo acolhimento à Turma MPROF2014Redes, competência e qualidade de ensino que são as marcas indeléveis do Cin/ UFPE.

Agradecimento especial ao meu orientador **Professor Doutor Carlos André Guimarães Ferraz (o Carlinhos)**, pela oportunidade em me aceitar como seu orientando e pela confiança em mim depositada.

Agradecer de forma fraterna à Secretária Acadêmica do Cin/ UFPE, a Senhora **Leila Oliveira de Azevedo e Silva**, que recebeu a todos os alunos como se fossem seus filhos e, os tratou com o devido respeito e, também, com a devida correição.

Aos **meus colegas** da turma MPROF2014Redes, em especial aos **meus Irmãos André Macêdo, Brainer Oliveira, Bruno Henrique, Chirlando Rocha, David Cunha, Giovani Jahn, Jadson Júnior e Janderson Souza**, pelo companheirismo, apoio, discussões, sessões de estudos, momentos de descontração, de diversão e de comemorações em suas companhias.

À **cidade do Recife**, que nos recebeu sempre de braços abertos e que, com o carinho de terra natal, acolheu a todos nós forasteiros.

Ao **Instituto Federal do Amapá (IFAP)** pelo apoio financeiro para o cumprimento de todas as etapas do Curso de Mestrado.

E, finalmente, agradecer **a todas as pessoas** que de uma forma positiva ou não, me incentivaram e me impulsionaram a alcançar mais um degrau em minha carreira acadêmica e profissional.

Obrigado a todos!

*“Para realizar grandes conquistas, devemos não apenas AGIR,
mas também SONHAR; não apenas planejar, mas também
ACREDITAR”*

— Jacques Anatole François Thibault

RESUMO

A análise do registro de eventos – conhecido como *logs* – em equipamentos e sistemas computacionais é útil para a identificação de atacantes, para delinear o modo de ataque e, também, para identificar quais são as falhas que foram exploradas. Há alguns trabalhos e pesquisas que demonstram a vantagem para uma política de segurança da informação em manter os *logs* de comunicação em redes e sistemas computacionais, focando nas trilhas para auditoria, que é um conjunto de ações onde se inclui: a coleta, o armazenamento e o tratamento dos *logs* de equipamentos, de aplicações e de sistemas operacionais, dentro de um sistema de computação. A auditoria de eventos atua na coleta de elementos que possam individualizar comportamentos perigosos de usuários, internos ou externos, ou eventos de sistema que possam vir a comprometer o uso aceitável dos recursos computacionais ou a quebra da tríade da segurança da informação: a confidencialidade, a integridade e a disponibilidade. Percebe-se que, dentro do modelo de política de segurança adotado nos centros de operação (*datacenters*) dos Institutos Federais de Educação, em sua imensa maioria, não utilizam o recurso de servidor de *logs*, ou tão somente atêm-se ao registro de eventos em seus sistemas e *hosts*, de forma individual e, sem o contexto da centralização e do tratamento dos registros dos eventos. A coleta dos *logs*, na modalidade *loghost* centralizado guarda, em um único repositório, os registros de eventos de diversos sistemas, equipamentos e serviços de rede, o que possibilitará uma análise do montante de *logs* adquiridos e a possibilidade de gerar trilhas de comportamento de usuários, além de permitir o cruzamento de informações conexas à autenticação de usuários. O recurso que permite a comunicação entre as aplicações, os sistemas operacionais e os equipamentos de rede, informando os eventos a serem registrados é o protocolo *Syslog* (*system log*), que é um padrão criado pela IETF para a transmissão de mensagens de *log* em redes IP. O objetivo maior deste trabalho é estabelecer um modelo para a análise e auditoria de *logs*, com o fim de identificar ações de usuários em uma rede com servidor de autenticação, aplicando a extração de informações úteis para o Gerenciamento da Segurança, elemento este levado a execução via o uso de *scripts* no formato PS1 (*Windows PowerShell*), atuando sobre os arquivos de *logs* dos Eventos de Segurança (*Security.evtx*) e gerando relatórios dos eventos relativos ao serviço de autenticação (*Active Directory*). Ressaltando que as funções implementadas pelos *scripts* não são disponibilizadas nativamente pelos sistemas *Windows* e, que o ferramental desenvolvido é de grande valor às atividades diárias do Administrador de Redes, concluindo-se que esta pesquisa apresenta um modelo de análise de *logs* para auditoria de registro de eventos.

Palavras-chaves: Auditoria de *logs*. *Loghost* centralizado. *Log* de eventos. Protocolo *Syslog*. Trilhas para auditoria.

ABSTRACT

The analysis of the events log - known as logs - in equipment and computer systems, is useful for the attackers' identification, to delineate the attack way and also to identify which faults have been exploited. There are some papers and researches that demonstrate the advantage of an information security policy in maintaining communication logs in networks and computer systems, focusing on the audit trails, which is a set of actions, which includes: collection, storage and the treatment of equipment logs, applications, operating systems, within a computer system. The audit of events, acts in the collection of elements that can individualize users dangerous behaviors, internal or external, or system events that may compromise the acceptable use of computing resources or the breakdown of the triad of information security: confidentiality, integrity and availability. It's noticed that, in the security policy model, adopted in the operation centers (datacenters) at the Federal Institutes of Education, in their vast majority, they do not use the log server resource, or only they attend the record of events, in their systems and hosts, individually and without the context of centralization and processing of event logs. The collection of logs, in the host-based mode, stores in a single repository, the event logs of various systems, equipment and network services, which will allow an analysis of the amount of logs acquired, the possibility of generating user behavior trails and, the crossing of information related to user authentication. The resource for communication between applications, operating systems and network equipment, informing the events to be registered is the Syslog protocol (system log), it's a standard created by the IETF, for the transmission of log messages in IP networks. The main goal of this work is to establish a model for the analysis and audit of logs, in order to identify actions of users in a network with authentication server, applying the extraction of useful information to the Security Management, element this led to execution through the use of scripts in the PS1 (Windows PowerShell) format, by acting on the Security Event log files (Security.evtx) and generating reports of events related to the authentication service (Active Directory). Note that the functions implemented by the scripts are not made available natively by Windows systems, and that the tooling developed is of great value to the daily activities of the Network Manager, and it is concluded that this research presents a log analysis model for event log audit.

Keywords: Audit of logs. Audit trail. Event Log. Host-based log collector. Syslog protocol.

LISTA DE FIGURAS

Figura 1: Diretorias sistêmicas – organograma da diretoria de TIC/ IFAP	19
Figura 2: Representação das camadas do protocolo <i>Syslog</i>	34
Figura 3: Modelo de captura de <i>logs</i> : dispositivo – coletor	38
Figura 4: Modelo de captura de <i>logs</i> : dispositivo – encaminhador – coletor	39
Figura 5: Mapa dos serviços de rede no IFAP	48
Figura 6: Ferramenta Usuários e Computadores do <i>Active Directory</i>	52
Figura 7: Confirmação da adição da máquina coletora de eventos	53
Figura 8: Adicionar a estação coletora de eventos ao grupo Administradores	54
Figura 9: Configuração das diretivas de grupo – ativar <i>log</i> de eventos	55
Figura 10: Habilitar os registros de auditoria – êxito e falha	56
Figura 11: Diretivas para o registro dos eventos de auditoria e de segurança	56
Figura 12: Visualizador de eventos no servidor origem – Eventos de Segurança	57
Figura 13: Confirmação da ativação da coleta dos <i>logs</i> de Segurança	59
Figura 14: Criar assinatura no visualizador de eventos da estação coletora	60
Figura 15: Início do Serviço Coletor de Eventos, com inicialização automática	61
Figura 16: Configurar uma assinatura no Visualizador de Eventos	62
Figura 17: Adicionar computadores do Domínio	62
Figura 18: Selecionar computador origem dos eventos – assinatura de coleta	63
Figura 19: Teste de conectividade com o computador origem dos eventos	63
Figura 20: Criar um filtro dos <i>logs</i> de Eventos no coletor	64
Figura 21: Seleção dos filtros dos <i>logs</i> a serem coletados do servidor origem	65
Figura 22: Lista das assinaturas criadas para a coleta de eventos	66
Figura 23: Visualizador de Eventos na estação coletora – <i>logs</i> de Segurança	66
Figura 24: Modelo aplicado à coleta dos <i>logs</i> – <i>loghost</i> centralizado	67
Figura 25: Consulta SQL ao arquivo de <i>logs</i> , na ferramenta de linha de comandos	70
Figura 26: Tela inicial da ferramenta <i>Microsoft Log Parser Studio 2.0</i>	70
Figura 27: Seção de montagem das <i>queries</i> no <i>Log Parse Studio 2.0</i>	71
Figura 28: Modelo proposto para a análise e auditoria dos registros de eventos	72
Figura 29: Abstração do ambiente real de implantação da captura de <i>logs</i>	74
Figura 30: Recursos de processamento da estação coletora	75
Figura 31: Arquivos rotacionados dos registros de eventos, no servidor origem	76
Figura 32: Arquivos rotacionados dos registros de eventos, na estação coletora	77
Figura 33: Ferramenta <i>Log Parser Studio 2.0</i> – análise em lotes de arquivos de eventos	78

Figura 34: <i>Query</i> – todos os eventos ocorridos nos registros coletados	79
Figura 35: Relação de todos os eventos ocorridos nos registros coletados	79
Figura 36: Gráfico dos tipos de eventos <i>versus</i> o número de ocorrências.....	80
Figura 37: <i>Query</i> 01 – encontrar os eventos 4624 – <i>logons</i> via RDP.....	81
Figura 38: Arquivo de saída do resultado da consulta aos ID 4624	81
Figura 39: <i>Query</i> 02 – encontrar os eventos 4625 – erros nas tentativas de <i>logon</i>	82
Figura 40: Arquivo de saída do resultado da consulta aos ID 4625	82
Figura 41: <i>Query</i> 03 – encontrar os eventos 4648 – <i>logon</i> com credenciais explícitas.....	83
Figura 42: Arquivo de saída do resultado da consulta aos ID 4648	84
Figura 43: <i>Query</i> 04 – encontrar os eventos 4660 e 4663 – exclusão de arquivos.....	84
Figura 44: Arquivo de saída do resultado da consulta aos IDs 4660 e 4663	85
Figura 45: <i>Query</i> 05 – encontrar os eventos 4771 – contas bloqueadas.....	85
Figura 46: Arquivo de saída do resultado da consulta aos IDs 4771.....	86
Figura 47: Aplicação do <i>script</i> Evento4625.ps1 no servidor em produção	87
Figura 48: Arquivo de saída da consulta <i>online</i> ao Evento <i>Security</i> – ID 4625	88
Figura 49: Informações constantes no campo <i>Strings</i> de um arquivo EVTX.....	91

LISTA DE QUADROS

Quadro 1: Formato da mensagem do protocolo <i>Syslog</i>	35
Quadro 2: Operação de cálculo da prioridade do protocolo <i>Syslog</i>	35
Quadro 3: Verificação do estado do serviço WinRM no sistema <i>Windows Server 2012</i>	51
Quadro 4: Adicionar máquina coletora ao grupo de leitores de eventos no servidor	52
Quadro 5: Situação inicial do SID do registro dos Eventos de Segurança	58
Quadro 6: Inclusão do SID da conta de Serviço de Rede ao canal de acesso	58
Quadro 7: Verificação do SID da conta de Serviço de Rede no canal de acesso	59
Quadro 8: Inicializar o serviço coletor de eventos do <i>Windows</i> , na estação coletora	61

LISTA DE TABELAS

Tabela 1: Descrição da metodologia da pesquisa	21
Tabela 2: Níveis e funções das camadas de um sistema de gerenciamento de <i>logs</i>	31
Tabela 3: Comparação das funcionalidades entre sistemas SIEM e gerenciamento de <i>logs</i>	33
Tabela 4: Camadas do protocolo <i>Syslog</i>	34
Tabela 5: Mensagens de facilidade do protocolo <i>Syslog</i>	35
Tabela 6: Níveis de severidade da mensagem <i>Syslog</i>	36
Tabela 7: Níveis de registro das mensagens de <i>logs</i>	37
Tabela 8: Nível do evento versus nível de configuração do registrador de eventos	38
Tabela 9: Comparação das propostas de métodos de análise de <i>logs</i>	46
Tabela 10: Eventos em análise nos registros dos <i>Logs</i> de Segurança	68
Tabela 11: Estrutura de arquivo de <i>log</i> de eventos nos sistemas <i>Windows</i> (EVTX)	68

LISTA DE ACRÔNIMOS

ABNT	Associação Brasileira de Normas Técnicas
AD	<i>Active Directory</i> – Diretório Ativo
BI	<i>Business Intelligence</i> – Inteligência de Negócios
CAFe	Comunidade Acadêmica Federada
CERT.BR	Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil
CLI	<i>Command Line Interface</i> – Interface de Linha de Comando
CORI	Coordenação de Redes e Infraestrutura
CSV	<i>Comma Separated Values</i> – Valores Separados por vírgula
DHCP	<i>Dynamic Host Configuration Protocol</i> – Protocolo de configuração dinâmica de hosts
DITI	Diretoria de Tecnologia da Informação
DNS	<i>Domain Name System</i> – Sistema de Nomes de Domínio
DTMF	<i>Distributed Management Task Force</i> – Grupo de Trabalho de Gerenciamento Distribuído
EVTX	<i>Event Log File XML format</i> – Arquivo de Eventos no formato XML
FTP	<i>File Transfer Protocol</i> – Protocolo para a Transferência de Arquivos
GPO	<i>Group Policy Object</i> – Objeto de Política de Grupo
GUI	<i>Graphical User Interface</i> – Interface Gráfica de Usuário
HTTPS	<i>Hyper Text Transfer Protocol Secure</i> – Protocolo de Transferência de Hipertexto Seguro
IEC	<i>International Electrotechnical Commission</i> – Comissão Eletrotécnica Internacional
IETF	<i>Internet Engineering Task Force</i> – Grupo de Trabalho de Engenharia da Internet
IFAP	Instituto Federal de educação, ciência e tecnologia do Amapá

IFs	Institutos Federais
IP	<i>Internet Protocol</i> – Protocolo de <i>Internet</i>
ISO	<i>International Organization for Standardization</i> – Organização Internacional para Padronização
KDD	<i>Knowledge Discovery in Databases</i> – Descoberta de Conhecimento em Bases de Dados
LMS	<i>Log Management System</i> – Sistema de Gerenciamento de Registros
LPS	<i>Log Parse Studio</i> – Estúdio Analisador de Registros
NBR	Norma Brasileira
NBSO	NIC.BR <i>Security Office</i> – Gabinete de Segurança do NIC.BR
NIC.br	Núcleo de Informação e Coordenação do Ponto BR
NIST	<i>National Institute of Standards and Technology</i> – Instituto Nacional de Padrões e Tecnologia
NTP	<i>Network Time Protocol</i> – Protocolo de Tempo para Redes
PDC	<i>Primary Domain Controller</i> – Primeiro Controlador do Domínio
PROAD	Pró-reitoria de Administração
RDP	<i>Remote Desktop Connection</i> – Conexão de Área de Trabalho Remota
RFC	<i>Request For Comments</i> – Requisição para Comentários
RNP	Rede Nacional de Pacotes
SAL	<i>Security Analytics Lab</i> – Laboratório de Análise de Segurança
SDDL	<i>Security Descriptor Definition Language</i> – Linguagem de Definição do Descritor de Segurança
SID	<i>Security Identifier</i> – Identificador Seguro
SIEM	<i>Security Information and Event Management</i> – Sistema de Gerenciamento de Informações de Segurança e Eventos
SNMP	<i>Simple Network Management Protocol</i> – Protocolo Simples para o Gerenciamento de Redes

SQL	<i>Structured Query Language</i> – Linguagem de Consulta Estruturada
SYSLOG	<i>System Log</i> – Registro de Sistema
TIC	Tecnologia da Informação e Comunicação
TXT	<i>Plain Text File</i> – Arquivo de Texto Simples
UDP	<i>User Datagram Protocolo</i> – Protocolo de Datagrama do Usuário
UUCP	<i>Unix-to-Unix Copy Protocol</i> – Protocolo de cópia Unix-para-Unix
UTC	<i>Universal Time Coordinated</i> – Horário Universal Coordenado
VoIP	<i>Voice over Internet Protocol</i> – Voz sobre Protocolo da Internet
WinRM	<i>Windows Remote Management</i> – Serviço de Gerenciamento Remoto do Windows
WMF	<i>Windows Management Framework</i> – Modelo de Gerenciamento do Windows
WS-Eventing	<i>Web Services Eventing</i> – Eventos baseados em Serviços da Web
WS-Man	<i>Web Services Management</i> – Gerenciamento de Serviços Web
XML	<i>eXtensible Markup Language</i> – Linguagem de Marcação Extensível

SUMÁRIO

1	INTRODUÇÃO	18
1.1	Motivação e Justificativa	18
1.2	Objetivos da Pesquisa	20
1.2.1	Objetivo Geral	20
1.2.2	Objetivos Específicos	20
1.3	Aspectos Metodológicos	21
1.4	Estrutura da Dissertação	22
2	FUNDAMENTAÇÃO TEÓRICA	24
2.1	Trilhas para Auditoria da Segurança da Informação	24
2.1.1	As Informações que devem ser Coletadas nos <i>Logs</i>	25
2.1.2	O processo de Coleta dos <i>Logs</i>	25
2.1.3	O Volume de Coleta dos <i>Logs</i>	25
2.1.4	Manipulação e Preservação de Dados para Auditoria.....	26
2.1.5	Considerações Legais sobre o Registro de Eventos	26
2.1.6	A Auditoria dos Registros de Eventos.....	26
2.2	Registro de Eventos de Sistema (<i>Logs</i>)	28
2.2.1	A Definição do que são os <i>Logs</i>	28
2.2.2	A Importância do Registro de <i>Logs</i>	28
2.2.3	A Geração de <i>Logs</i>	29
2.2.4	O Armazenamento de <i>Logs</i>	29
2.2.5	O Monitoramento de <i>Logs</i>	30
2.2.6	O Gerenciamento de <i>Logs</i>	30
2.2.7	Sistema de Gerenciamento de Informações de Segurança e Eventos (SIEM)	31
2.3	A Infraestrutura para a Retenção e o Tratamento de <i>Logs</i>	33
2.3.1	O Protocolo <i>Syslog</i>	33
2.3.2	O Padrão de Mensagens do Protocolo <i>Syslog</i>	34
2.3.3	Serviço de Coleta de <i>Logs</i> – Modelo <i>Loghost</i> Centralizado	38
2.3.4	Rotacionamento de <i>Logs</i>	39

3	TRABALHOS RELACIONADOS	41
3.1	Análise, Mineração e Auditoria de <i>Logs</i>	41
3.2	Gerenciamento e Correlação de Eventos de Segurança	43
3.3	Análise de <i>Logs</i> de Autenticação em Rede	44
3.4	Comparação entre as Propostas de Análise de <i>Logs</i>	45
4	MODELO DE ANÁLISE DE <i>LOGS</i> PARA AUDITORIA DE REGISTRO DE EVENTOS	47
4.1	Cenário para o Estudo de Caso	47
4.1.1	Configuração do Servidor Origem dos <i>Logs</i> – Windows Server 2012 (AD)	49
4.1.2	Configuração da Estação Coletora de <i>Logs</i> – Windows 7 Professional SP1	59
4.2	Método Proposto para a Análise e Auditoria dos Eventos de Segurança	67
5	ANÁLISE DOS RESULTADOS	74
5.1	Ambiente de Testes	74
5.1.1	Detalhes das Configurações	75
5.2	Aplicando a Técnica Adotada para a Análise e Auditoria de <i>Logs</i>	77
5.3	Resultados Alcançados	86
6	CONCLUSÃO E TRABALHOS FUTUROS	89
6.1	Conclusão	89
6.2	Dificuldades Encontradas	90
6.3	Trabalhos Futuros	91
	REFERÊNCIAS	92
	APÊNDICES	96
	A - Query – Todos os Eventos Ocorridos nos Registros Coletados	97
	B - Query 01 – Encontrar os Eventos 4624 – <i>Logons</i> Via RDP	98
	C - Script PS1 – Encontrar os Eventos 4624	99
	D - Query 02 – Encontrar os Eventos 4625 – Erros nas Tentativas de <i>Logon</i>	101
	E - Script PS1 – Encontrar os Eventos 4625	102
	F - Query 03 – Encontrar os Eventos 4648 – <i>Logon</i> com Credenciais Explícitas	104
	G - Script PS1 – Encontrar os Eventos 4648	105
	H - Query 04 – Encontrar os Eventos 4660 E 4663 – Exclusão de Arquivos	107
	I - Script PS1 – Encontrar os Eventos 4660 E 4663	108
	J - Query 05 – Encontrar os Eventos 4771 – Contas Bloqueadas	110
	K - Script PS1 – Encontrar os Eventos 4771	111

1

INTRODUÇÃO

Neste capítulo são apresentadas as motivações e justificativas que nortearam a realização deste trabalho. Também são descritos os objetivos, geral e específicos seguidos de uma breve exposição das condições necessárias para o cumprimento dos mesmos. Ao final, o Capítulo é encerrado com uma seção que expõe a estrutura utilizada para construção da pesquisa.

1.1 Motivação e Justificativa

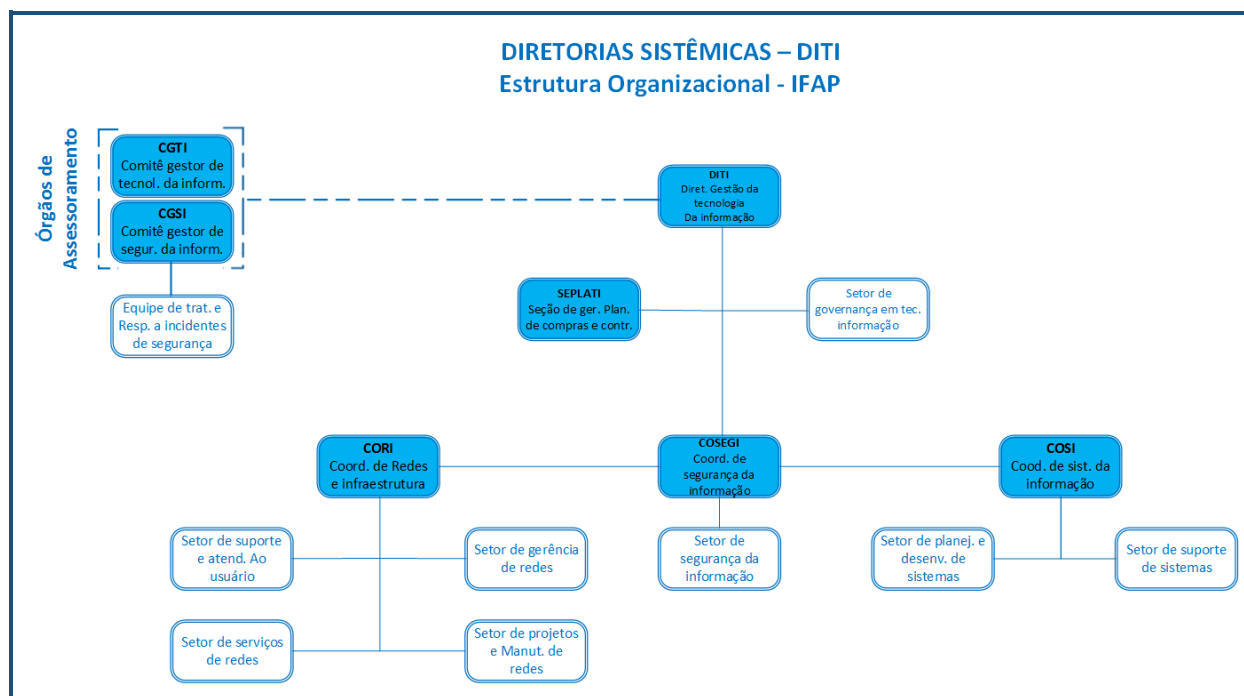
No levantamento sobre segurança cibernética da (PwC, 2016) conclui-se que os ataques às redes e sistemas computacionais, têm se intensificado nos últimos anos e estão sendo utilizados como ferramenta de extorsão monetária, vazamento de informações sensíveis, como ferramenta de pressão política e, também, como forma de protesto ideológico. Neste ínterim, é fato frequente, que as invasões de sistemas computacionais promovem acesso à informações e recursos de forma não autorizada, trazendo acentuada preocupação para os gestores de recursos e segurança de Tecnologia da Informação e Comunicação (TIC). Tais ações recaem não somente a sistemas de usuário doméstico, mas peculiarmente inserem-se em cenários de redes de computadores mais sofisticados, dos quais se esperava uma estrutura mais firme de segurança. O aumento dos incidentes de segurança da informação, conforme (PwC, 2016) são estarrecedores, uma vez que se registra ano a ano o crescimento vertiginoso dos ataques cibernéticos, cada vez demonstrando um aumento de gravidade impacto aos recursos computacionais das organizações. E, ainda, (PwC, 2016) chama a atenção que os atuais modelos de prevenção e detecção se mostraram, bastante ineficazes em relação aos ataques que são cada vez mais sofisticados, ressalta-se ainda que as organizações, ainda, não têm a exata noção do que fazer e nem conhecem quais são os recursos necessários a serem empregados para o combate aos atacantes que são altamente qualificados e agressivos.

Em alusão à premente necessidade de se buscar a apuração dos fatos corridos em uma invasão, temos a recomendação de (SON; *et al.*, 2015), onde afirma que mesmo após um determinado evento de violação de segurança ter sido perpetrado e a equipe de resposta a incidentes registrar o fato ocorrido, há que se buscar a identificação do tipo de incidente, além de identificar os meios utilizados para o ataque, ou ainda se possível, identificar o ente atacante. É

pertinente que se tenha os registros de eventos ocorridos diariamente e, manter-se esses registros em local seguro e disponibilizar sobre estes uma posterior análise que comprove a fonte da vulnerabilidade, objetivando prover uma mudança sistemática com a finalidade de sanar as vulnerabilidades e livrar-se do alcance de novas ameaças. No atual cenário das redes de computadores, há de se que fundir as funções de gestão de riscos e gestão de segurança no ambiente corporativo.

Tomando-se como base a estrutura de Tecnologia da Informação, adotada pelos Institutos Federais de Educação (IFs) de todo o Brasil, se constata que há um modelo de organograma, para o setorial de Tecnologia da Informação, adotado por todos os IFs, tornando-os semelhantes em suas divisões de unidades gerenciais e operacionais, conforme a Figura 1.

Figura 1: Diretorias sistêmicas – organograma da diretoria de TIC/ IFAP



Fonte: Pró-reitoria de Administração (PROAD/ IFAP, 2017)

Este modelo de organograma aborda a divisão de funcionalidades por áreas de competências críticas, com base nos serviços ofertados pela Diretoria de Tecnologia da Informação do Instituto Federal de educação, ciência e tecnologia do Amapá (DITI/ IFAP). E tomando-se a Coordenação de Redes e Infraestrutura (CORI/ DITI) como a fonte de atuação para os serviços e sistemas estruturantes da TIC e analisando os serviços que essa divisão setorial oferece, percebe-se a inerente necessidade de alocarmos um serviço próprio para a retenção e, caso seja plausível, o tratamento dos registros e das trilhas para auditoria. O modelo que será adotado no cenário de testes é a coleta de *logs* de servidor de serviço de rede, no caso em tela de servidor de autenticação baseado na solução AD (*Active Directory*) e, posterior uso de ferramenta especialista para analisar os tipos de *logs* capturados. E como forma de armazenamento de *logs*,

será utilizado o modelo denominado *loghost* centralizado, que é um modelo de armazenamento de eventos em sistema dedicado para a coleta e ao armazenamento de *logs* de outros sistemas em rede, e este sistema dedicado de coleta atuará como um repositório redundante de *logs*. Nesse modelo de armazenamento de eventos – *loghost* centralizado – não há a disponibilização de nenhum outro serviço de rede, nem acesso remoto para os administradores, com o fim de minimizar a possibilidade de comprometimento da segurança do sistema de coleta de eventos e, essa restrição se reverte como vantagem de segurança da informação, uma vez que esse modelo facilita a análise dos *logs* e a correlação de eventos ocorridos em sistemas distintos (NIC.BR, 2003).

1.2 Objetivos da Pesquisa

Nesta seção serão descritos os objetivos: geral e específicos, os quais se busca alcançar por meio deste trabalho de pesquisa.

1.2.1 Objetivo Geral

O objetivo maior deste trabalho é propor um modelo para a análise e auditoria de *logs*, no qual seja possível a consulta dos registros dos Eventos de Segurança, com base em regras pré-configuradas para a análise de informações referentes ao Serviço de Autenticação (*Active Directory*) utilizando *scripts Windows PowerShell*, aplicáveis em ambientes de rede de pequeno e médio porte.

1.2.2 Objetivos Específicos

- Elencar os conceitos e recursos que compõem a infraestrutura e que permitirão a coleta e retenção de *logs* em repositório distinto da origem dos eventos;
- Identificar as abordagens de a análise e auditoria de registro de eventos com o objetivo de propor um modelo, baseado em práticas já consolidadas;
- Implementar um ambiente para a análise de *logs*, coletando os registros de Eventos de Segurança provenientes do sistema *Windows Server*, com o fim auditar os eventos conexos ao serviço de autenticação (*Active Directory*);
- Aplicar um modelo para a análise e auditoria de *logs* dos eventos de autenticação de usuários e serviços, utilizando *Scripts Windows PowerShell* com o fim de consultar eventos que sejam conexos ao Serviço de Autenticação (*Active Directory*).

1.3 Aspectos Metodológicos

Presume-se o alcance dos objetivos supracitados, através de pesquisa literária, fundamentada em revisão da literatura, em primeiro momento, acerca dos métodos atualmente utilizados para análise de *logs*. E, posteriormente, sugere-se uma efetiva e criteriosa atividade de análise sobre *logs* utilizando duas ferramentas: *Log Parser 2.2* e *Log Parser Studio 2.0*, a primeira operando apenas no ambiente de linha de comandos (CLI) e a segunda voltada para o ambiente gráfico (GUI), ambas operam no sistema *Windows*. E para teste comparativo na análise dos *logs*, a base estrutural para estas ações está a cargo de um servidor encaminhador de *logs*, baseados em sistema operacional *Windows Server 2012*; uma estação de trabalho utilizando sistema operacional *Windows 7 Professional SP1*, que atuará com a função de estação coletora, no modelo *loghost* centralizado.

Para (GIL, 2002) os trabalhos de pesquisa científica classificam-se com base na análise aos seus pontos de características metodológicas e, que as pesquisas científicas são classificadas sob quatro diferentes perspectivas, as quais sejam:

- De acordo com a natureza da pesquisa: pesquisa básica ou aplicada;
- Sobre a forma de abordagem do problema: pesquisa quantitativa ou qualitativa;
- Sob a perspectiva dos objetivos do estudo: pesquisa exploratória, descritiva ou explicativa;
- A partir dos procedimentos técnicos a serem adotados: pesquisa bibliográfica, documental, experimental, levantamento, estudo de caso, ação, participante.

Então, dentro destas quatro perspectivas denota-se que esta pesquisa está classificada metodologicamente conforme a Tabela 1.

Tabela 1: Descrição da metodologia da pesquisa

descrição da metodologia	
NATUREZA DA PESQUISA	APLICADA
ABORDAGEM DO PROBLEMA	QUALITATIVA
PERSPECTIVA DOS OBJETIVOS	EXPLORATÓRIA
PROCEDIMENTOS TÉCNICOS	ESTUDO DE CASO

Fonte: Elaborado pelo autor

Quanto à natureza da pesquisa, classifica-se como pesquisa aplicada, pois o objetivo deste trabalho é gerar conhecimentos de aplicação prática, direcionados à solução de problemas específicos, envolvendo verdades e interesses restritos aos ambientes semelhantes ao da pesquisa.

Quanto à abordagem do problema, a pesquisa tem a característica qualitativa, pois o desenvolvimento da pesquisa é imprevisível, o conhecimento do pesquisador é parcial e limitado e, o objetivo do trabalho é o de produzir informações aprofundadas e ilustrativas.

Quanto à perspectiva dos objetivos, essa pesquisa define-se por exploratória, pois esta pesquisa tem o objetivo de proporcionar maior familiaridade com o problema, com a intenção de torná-lo mais explícito ou a construir hipóteses, inclui também o levantamento bibliográfico sobre o tema pesquisado, dando subsídios ao pesquisador de formar seu arcabouço de estudos e direcioná-lo a um resultado esperado.

Quanto aos procedimentos técnicos a pesquisa caracteriza-se como um estudo de caso, tendo em vista que a situação foi ampla e exaustivamente estudada, para se obter uma compreensão a respeito do problema e, a maneira de execução da pesquisa pode ser aplicada às situações parecidas com situação desta pesquisa. Mas o seu principal objetivo é descrever detalhadamente, no caso do trabalho proposto, uma forma de análise de *logs* com o fim de encontrar eventos relevantes para o Gerenciamento da Segurança em um ambiente de rede, ocorrendo a aplicação desta pesquisa sobre os dados reais de *logs* extraídos do serviço *Active Directory* do IFAP e, sendo fornecido à Coordenação de Redes e Infraestrutura (CORI/ IFAP) o ferramental em *scripts* PS1, resultante das interações e atendendo a proposta de análise e auditoria da pesquisa.

1.4 Estrutura da Dissertação

Este trabalho está organizado da seguinte forma: No primeiro capítulo se apresenta uma introdução do trabalho, com a finalidade dar início ao entendimento e à abordagem do tema durante todo o trabalho e, neste mesmo capítulo estão descritos a motivação e a justificativa e, também os objetivos, geral e específicos desta dissertação.

No **Capítulo 2**, há a apresentação dos conceitos correlacionados com a pesquisa, abordando as tecnologias, conceitos e protocolos envolvidos na aquisição de trilhas para auditoria, focando na retenção de *log* de eventos, delineando a importância de se ter a ação positiva, em um ambiente corporativo, de reter os *logs* de eventos, abordando os recursos necessários para a geração, o armazenamento e o monitoramento de *logs*. E, ainda, neste capítulo, realiza-se a análise dos trabalhos relacionados com o tema central da pesquisa, abordando as técnicas já consolidadas para a auditoria de eventos, apresentando três técnicas estudadas.

No **Capítulo 3**, busca-se os trabalhos relacionados ao tema da pesquisa, para se encontrar o estado da arte no que se refere aos temas de análise de *logs*; mineração de *logs* e auditoria de *logs*, identificando-se a técnica que será adotada como embasamento para o desenvolvimento do

ponto central deste trabalho e, ainda, apresenta-se um comparativo entre os trabalhos pesquisados, demonstrando o diferencial do método propostos nesta pesquisa.

No **Capítulo 4**, o trabalho se debruça sobre o tema central da pesquisa, abordando um modelo-base de referência para a configuração de encaminhamento de *logs* de 01 servidor *Windows Server 2012*, este servidor tem a função de primeiro controlador de domínio (*Primary Domain Controller – PDC*) e; 01 estação de coletora de eventos, no modelo *loghost* centralizado, implementada em sistema operacional *Windows 7 Professional SP1* para a coleta e armazenamento dos *logs*; 01 ferramenta *Log Parse Studio 2.0*, utilizada para a análise de *logs*, tendo como o foco do trabalho a análise dos *logs* dos Eventos de Segurança. O trabalho é embasado na prática do redirecionamento de *logs* de eventos do servidor de autenticação (*Active Directory*) para uma estação de coleta de eventos. E como quesito de conclusão, neste capítulo, será possível entregar um modelo de análise de *logs* dos Eventos de Segurança em um ambiente *Windows*, com o fim de subsidiar com ferramental intuitivo, o processo de auditoria e as atividades de Gerenciamento da Segurança em uma Rede de computadores.

No **Capítulo 5**, que é a análise dos resultados da pesquisa, aponta-se as situações em que há vantagem na adoção do modelo sugerido e, recomenda-se a adoção em outras unidades da Rede Federada (IFs), pois aquelas unidades possuem, com bastante proximidade, as mesmas estruturas de redes que do IFAP e reforça-se que a solução proposta foi desenvolvida com base nos testes e comparações das ferramentas *Log Parser 2.2* e *Log Parser Studio 2.0* que são ferramentas próprias da empresa *Microsoft* para os sistemas *Windows*. O objetivo é aprimorar a Gestão de Segurança da Informação, dentro das unidades da TIC dos Institutos Federais de Educação, ressaltando que dentre as ferramentas e soluções adotadas na segurança da informação, uma das últimas coisas lembradas é a retenção e o tratamento de *logs*, que servirá de trilhas para processos de auditoria de possíveis incidentes ou eventos que impactem nos serviços e recursos computacionais. E, encerrando esse trabalho, **no Capítulo 6**, apresenta-se as conclusões da abordagem proposta neste trabalho, as dificuldades encontradas, ressaltando que algumas persistem e, algumas sugestões para o desenvolvimento de trabalhos futuros sobre o tema.

2

FUNDAMENTAÇÃO TEÓRICA

Neste capítulo estão elencados os trabalhos que serviram de base para os estudos e, para a afirmação do modelo-base proposto, as visões teóricas e, algumas recomendações do estado da arte para o processo de auditoria de *logs*. E, toda essa base teórica é o guia de orientação deste trabalho de pesquisa e, tem o fim de sustentar a abordagem adotada no desenvolvimento deste trabalho.

2.1 Trilhas para Auditoria da Segurança da Informação

De acordo com a RFC 2196 ([FRASER; et al., 1997](#)), uma política de segurança consiste num conjunto formal de regras que devem ser seguidas pelos utilizadores dos recursos de uma organização. As políticas de segurança devem ter implementação realista, e definir claramente as áreas de responsabilidade dos utilizadores, do pessoal de gestão de sistemas e redes e da direção. Deve também adaptar-se às alterações na organização. As políticas de segurança fornecem um enquadramento para a implementação de mecanismos de segurança, definem procedimentos de segurança adequados, processos de auditoria à segurança e estabelecem uma base para procedimentos legais na sequência de ataques. Para os procedimentos de coleta e manuseio de trilhas de auditoria a, em seu capítulo de auditoria, estão elencados os procedimentos de recolha de dados gerados na rede, os quais podem ser úteis na análise da segurança de uma rede e responder a incidentes de segurança.

Como o preceituado pela ([MICROSOFT, 2017](#))¹, a análise dos eventos registrados nos *logs* se revestem em ferramenta básica para o monitoramento e o rastreamento de possíveis atividades de intrusão e, os Administradores de rede devem adotar o procedimento diário de análise dos dados dos arquivos de *logs*, buscando por eventos não corriqueiros ou eventos inesperados, devendo apurar as atividades suspeitas.

¹ Detecção de intrusão: <https://technet.microsoft.com/pt-br/library/dd459005.aspx>.

2.1.1 As Informações que devem ser Coletadas nos *Logs*

Na RFC 2196 (FRASER; *et al.*, 1997) orienta-se que, para a eficiência do procedimento de análise dos eventos, coletados pelos *logs*, o sistema de auditoria deve incluir: a) as tentativas de elevação do nível de segurança de qualquer usuário, de um processo ou de outra entidade na rede e, isso inclui as atividades de *login*² e *logout*³; b) o acesso com credenciais de super-usuário, ou o equivalente, em sistemas não-UNIX; c) a geração de *tickets* (bilhetes de acesso, para *Kerberos*, por exemplo) e, d) qualquer outra alteração no acesso ou *status* (condição de usuário). De forma abrangente, as informações que serão coletadas em um sistema que prevê a auditoria de eventos incluem: a) nome de usuário e nome de *host* (equipamento conectado à rede), para *login* e *logout*; b) direitos de acesso anteriores e novos, para uma alteração dos direitos de acesso e; c) um *timestamp*⁴. E como há uma abrangência enorme no que há a disposição para se coletar, como registro de eventos, logo, há muito mais informação útil que pode ser recolhida, dependendo do que o sistema disponibiliza e o quanto de espaço estará disponível para o armazenamento dessas informações.

2.1.2 O processo de Coleta dos *Logs*

Existem basicamente três maneiras de armazenar registros de auditoria, em conformidade com a RFC 2196 (FRASER; *et al.*, 1997): em um arquivo de leitura/ gravação em um *host*, em um dispositivo de gravação, por exemplo, um disco ótico ou uma unidade de fita especialmente configurada. Para cada um dos métodos de armazenamento descritos, há também a questão de se proteger o caminho entre o dispositivo que gera o *log* e o dispositivo de registro real, ou seja, o servidor de arquivos, a unidade de fita ou disco ótico e, o *host* que gerou o registro. Se esse caminho for comprometido, o procedimento de coleta pode ser interrompido, falsificado ou ambas as situações podem ocorrer.

2.1.3 O Volume de Coleta dos *Logs*

Seguindo a RFC 2196 (FRASER; *et al.*, 1997) a coleta de dados de auditoria pode resultar em enorme acumulação de *bytes*, portanto dispor de recursos de armazenamento para os registros coletados é algo que deve ser planejado. Há maneiras de se reduzir o espaço de armazenamento para grandes volumes de registros. A primeira maneira pode ser a compactação dos registros

² *Login* é o procedimento de ingresso em uma sessão de sistema, utilizando credenciais.

³ *Logout* é procedimento de desvincular-se de uma sessão de sistema.

⁴ *Timestamp*: registro digital de data e hora da ocorrência de um evento específico de sistema.

coletados, usando qualquer método de compactação. E a segunda maneira, é a restrição do espaço necessário para o armazenamento, mantendo-se os registros por um período mais curto de tempo e, com apenas resumos dos registros mantidos em outros arquivos de longo prazo. Um inconveniente deste último método, refere-se à resposta a incidentes, pois na ocorrência de um incidente, o Administrador do serviço começa a investigar o incidente em curso e, então nesse momento é de grande utilidade que o Administrador possa ter acesso aos registros detalhados para a correta auditoria e, se os registros forem apenas resumos, pode não haver detalhes suficientes para o tratamento correto do incidente.

2.1.4 Manipulação e Preservação de Dados para Auditoria

Os registros de auditoria devem ser o tipo de dados dos mais protegidos na estrutura de recursos computacionais, conforme orientação da RFC 2196 (FRASER; *et al.*, 1997), tanto no local de coleta quanto em seus armazenamentos de longa vida, pois se um intruso obtiver o acesso aos registros de auditoria ou aos próprios sistemas geradores de eventos, isto colocaria em risco todo o sistema de auditoria. Os registros de eventos, em casos específicos, vêm a ser o elemento chave para todo o processo de investigação, apreensão e acusação do autor de um incidente.

2.1.5 Considerações Legais sobre o Registro de Eventos

Os comprometimentos legais abordados na RFC 2196 (FRASER; *et al.*, 1997), reforçam que devido ao conteúdo dos registros de evento, há uma série de questões jurídicas que podem surgir e que precisam ser tratadas pelos órgãos jurídicos da Organização. Então, se existe formalmente definido o procedimento de coleta e armazenamento dos registros de eventos para auditoria, há a que se responsabilizar pelas consequências resultantes do armazenamento destes registros, tanto pela sua existência, quanto pelo seu conteúdo. Por esse motivo, é aconselhável solicitar ao departamento jurídico, que se decida sobre a abordagem dada aos dados coletados e mantidos e, como os mesmos devem ser tratados dentro da política corporativa da organização.

2.1.6 A Auditoria dos Registros de Eventos

Um sistema de auditoria, na pesquisa de (TSUNODA; *et al.*, 2009), é utilizado para verificar o registro de informações de um determinado evento, caso ocorra por exemplo, o acesso indevido por parte de usuários, ou ocorra algum incidente que necessite ser auditado. No contexto do procedimento de auditoria, são coletados dados para a verificação de onde partiu o problema de segurança e, a auditoria também visa conhecer quais foram as ações praticadas e, estabelecer o

padrão de ação dos suspeitos. As informações do registro de eventos são de grande valor para o Administrador de redes e de sistemas, especialmente para monitorar e gerenciar as operações da rede. E, a análise de *logs* atingiu outros campos do ambiente computacional, como o gerenciamento de segurança, a auditoria e a investigação forense. À medida que o escopo de abrangência do uso do registro de *logs* se expande, os requisitos para a coleta, armazenamento e retenção dos *logs* estão ficando mais rigorosos e complexos. A auditoria exige informações exaustivas não só dos servidores da rede, mas também de todos os sistemas em rede, incluindo os registros das estações dos usuários.

De acordo com a norma ([ABNT NBR ISO/ IEC 27002:2013](#)), ao controle convém que os registros de eventos para o processo de auditoria, os quais contenham atividades dos usuários, exceções e outros eventos de segurança da informação sejam produzidos e mantidos por um período de tempo acordado, com a alta gestão e o setor jurídico da organização, com o fim de auxiliar em futuras investigações e monitoramento do controle de acesso. E recomenda que nos registros de eventos, para o processo de auditoria, estejam incluídos quando relevantes as seguintes informações:

- Identificação dos usuários;
- Datas, horários e detalhes de eventos-chave, como, por exemplo, horário de entrada (*log-on*) e saída (*log-off*) no sistema;
- Identidade do terminal ou, quando possível, a sua localização;
- Registros das tentativas de acesso ao sistema aceitas e rejeitadas;
- Registros das tentativas de acesso a outros recursos e dados aceitos e rejeitados;
- Alterações na configuração do sistema;
- Uso de privilégios;
- Uso de aplicações e utilitários do sistema;
- Arquivos acessados e tipo de acesso;
- Endereços e protocolos de rede;
- Alarmes provocados pelo sistema de controle de acesso;
- Ativação e desativação dos sistemas de proteção, tais como sistemas de antivírus e sistemas de detecção de intrusos.

Nos registros de *logs* existem dados confidenciais, onde somente um usuário com o acesso privilegiado poderá acessar, assim mantendo a total privacidade sobre esses registros.

2.2 Registro de Eventos de Sistema (*Logs*)

Na pesquisa de (TSUNODA; KEENI, 2014) define-se os registros de eventos de sistema, com sendo as mensagens de *logs* as quais são geradas por sistemas operacionais, por aplicativos e por equipamentos de *hardware* e, nestas mensagens estão contidas informações importantes sobre a saúde e o funcionamento de um sistema, estas mensagens também são de grande importância para o Gerenciamento da Segurança, para a auditoria de equipamentos e sistemas e, para a análise forense em uma rede local (*Intranet*). Assim, um sistema de registro de eventos que gera, encaminha, coleta e armazena as mensagens de *log*, deve ser monitorado e gerenciado como todos os outros componentes da infraestrutura de TIC de uma organização, para garantir que este sistema esteja funcionando normalmente, ou seja, garantir que os *logs* estejam sendo coletados e arquivados conforme as premissas da segurança da informação.

2.2.1 A Definição do que são os *Logs*

Registro de eventos, em ambiente computacional, é a documentação produzida de forma automática ou iniciada a partir de um sistema ou *host*, com data e hora de eventos relevantes para um sistema em particular, definição dada por (DU; LI, 2016). E, ainda, na mesma pesquisa afirma-se que os registros de eventos de sistema têm sido frequentemente utilizados como um recurso valioso com vistas a melhorar a saúde e a estabilidade de um sistema computacional – *software* ou *hardware*. Um procedimento típico para a análise dos *logs* de um sistema é primeiro analisar os arquivos de *logs* desestruturados e, em seguida, aplicar a análise de dados nos dados estruturados resultantes. No contexto atual das redes de computadores, todos os equipamentos de *hardware*, os aplicativos e os sistemas de *software* produzem arquivos de *log* e, tratá-los é uma exigência para o entendimento das ocorrências que os geraram.

2.2.2 A Importância do Registro de *Logs*

Nas definições do (NIC.BR, 2003) o principal objetivo da gestão de segurança da TIC é ser proativa e, também, tomar as medidas e ações que tornem o mais difícil possível que algum ataque ou comportamento comprometa a rede de computadores e seus sistemas. E, essa proatividade exige que o administrador de redes e sistemas seja capaz de detectar as falhas reais em seu ambiente e, entender como essas falhas estão sendo exploradas e, nesse momento é onde os dados de *log* poderão, de forma efetiva, ajudar a identificar possíveis vulnerabilidades, com o fim de expor um ataque ou identificar o dano causado, será necessário analisar os eventos de *log* gerados e, ao coletar e analisar *logs*, você pode entender o que acontece no comportamento da rede e nos sistemas. Cada arquivo de *log* contém muitas informações que podem ser valiosas,

especialmente se houver uma ação diligente para analisá-las e, com a análise adequada desses dados registrados, pode-se identificar tentativas de invasão, equipamentos mal configurados e outras informações relevantes, a exemplo da autenticação de usuários, também será possível detectar o uso indevido do ambiente da TIC, ataques, exploração de vulnerabilidades, rastrear e auditar as ações executadas por um usuário, além de detectar problemas em *hardwares* ou em sistemas que estejam sendo registrados seus eventos.

2.2.3 A Geração de *Logs*

Para que os *logs* de um sistema sejam úteis para um administrador, conforme o (NIC.BR, 2003) eles devem estar com o horário sincronizado via protocolo de tempo para redes (*Network Time Protocol* – NTP), devidamente coordenados com o horário universal coordenado (*Universal Time Coordinated* – UTC) e esses registros devem ser tão detalhados quanto possível sem, no entanto, gerar dados em excesso. Informações especialmente úteis são aquelas relacionadas a eventos de rede, tais como conexões externas e registros de utilização de serviços, a exemplo de arquivos transferidos via FTP, acessos a páginas *Web*, tentativas de *login* sem sucesso, avisos de disco cheio, entre outros. Para registrar estas informações de eventos, é necessário configurar o sistema ou *hardware* de maneira apropriada. E há diferentes maneiras de se configurar a geração de *logs*, sendo procedimentos distintos para cada sistema ou equipamento, podendo-se habilitar o *logging* de informações em sistemas, *softwares* específicos como *firewalls*; servidores HTTP e outros equipamentos em rede.

2.2.4 O Armazenamento de *Logs*

Para o (NIC.BR, 2003), há duas formas para o armazenamento dos registros de eventos que são: *on-line* e *off-line*. Na forma *online* admite-se duas variações, a primeira variação do armazenamento *on-line* dos registros de eventos é quando os *logs* são gerados e armazenados nos próprios sistemas e equipamentos geradores dos eventos; a segunda variação de armazenamento *on-line*, é através do uso de sistema de *loghost* centralizado, que tem como premissa o uso de um *host* que atuará como coletor e repositório dos *logs* coletados de outros sistemas ou equipamentos na rede. Com o fim de preservar esse sistema centralizado de armazenamento de *logs* é altamente recomendável que o sistema *loghost* centralizado não ofereça quaisquer outros serviços, eliminando inclusive o recurso de acesso remoto aos seus Administradores, o que minimiza a possibilidade do comprometimento dos registros coletados, se tornando uma vantagem o fato de se ter um único local para o armazenamento do registro de eventos de sistemas distintos, possibilitando a realização de análises, e a correlação de eventos de diferentes sistemas, em um único local.

E seguindo as orientações do (NIC.BR, 2003), como regra de segurança, há a recomendação de que os registros de eventos não permaneçam armazenados por um longo período na forma *on-line*, pois o volume dos registros pode ocupar um enorme espaço em disco, impactando na necessidade contínua e infinita de espaço para o armazenamento e, uma estratégia de bastante eficaz para solucionar este problema é o armazenamento *off-line*. Uma estratégia eficiente é transferir, periodicamente, os *logs* do disco onde são reunidos, para dispositivos de armazenamento *off-line*, tais como fita magnética ou disco ótico e, é altamente recomendável que seja gerado um *checksum* criptográfico, a exemplo do MD5 ou SHA-1, dos *logs* que são armazenados *off-line*, devendo-se manter o *checksum* separado dos arquivos de *logs*, para que possa, em eventual necessidade, utilizar o *checksum*, para verificar a integridade dos arquivos de *log*. Os arquivos de *logs* armazenados *off-line* devem ser mantidos por um certo período de tempo, tempo esse definido na política de segurança da informação da Organização, pois estes arquivos de *log* podem ser necessários para subsidiar um procedimento de auditoria.

2.2.5 O Monitoramento de Logs

Na pesquisa de (LIM; SINGH; YAJNIK, 2008) estabelece que as técnicas de monitoramento de *logs* com o fim de identificar o comportamento de sistemas e de usuários têm hoje uma significativa importância, tornando a análise de *logs* de sistemas como ferramenta principal na segurança da informação. E a técnica elementar para determinar os comportamentos é a mineração dos eventos gerados pelo protocolo *Syslog*, com o fim de detectar e prever comportamentos e falhas em sistemas computacionais e em atividades de usuários.

As recomendações do (NIC.BR, 2003), para a geração, armazenamento e monitoramento de *logs* é que se estabeleça um sistema que contenha as funcionalidades básicas:

- habilitação do *logging* em sistemas e serviços;
- estabelecimento de um procedimento de armazenamento de *logs*;
- instalação e configuração de um *loghost* centralizado;
- estabelecimento de um procedimento de monitoramento de *logs*;
- instalação de ferramentas de monitoramento automatizado e de sumarização de *logs*.

2.2.6 O Gerenciamento de Logs

Em publicação especial emitida pelo NIST (*National Institute of Standards and Technology*) em sua *Special Publication* 800-92 (publicação especial), torna padrão que a análise de registros de eventos é benéfica para que se possa identificar os incidentes de segurança, as

violações de políticas, as atividades fraudulentas e os possíveis problemas operacionais (KENT; SOUPPAYA, 2006). E como o corroborado por (YUE; XIAOBIN; ZHENGQIU, 2008), o amplo uso dos sistemas de gerenciamento da informação, trouxe a necessidade de se proceder ao gerenciamento de *logs*, de forma mais dedicada. E, considerando o cenário atual dos centros de informação – *datacenters* – não é admissível que um ambiente computacional não possua o serviço de registro de eventos. O modelo de gerenciamento de *logs* tem um papel relevante em outros cenários além da segurança da informação, a exemplo, na resolução de problemas de aplicativos e na administração do sistema operacional. E em (TOMONO; *et al.*, 2009) define-se que um sistema de gerenciamento de *logs* é composto por várias camadas: a) uma camada para coletar e armazenar os *logs* reais, b) uma camada para o armazenamento dos *logs* reais, em um banco de dados e, c) uma camada para a análise de *logs* por meio de gráficos e gerar relatórios. As camadas de um sistema de gerenciamento de *log* estão ilustradas na Tabela 2:

Tabela 2: Níveis e funções das camadas de um sistema de gerenciamento de *logs*

níveis	funções	o que manipula/ gera
Nível 2	análise de <i>logs</i>	gráficos etc.
Nível 1	gerenciamento de <i>logs</i>	SGBD
Nível 0	coleta e armazenamento de <i>logs</i>	<i>logs</i> reais

Fonte: Adaptado de (TOMONO; *et al.*, 2009)

No contexto do gerenciamento dos *logs*, (CHUVAKIN, 2016), reforça que um exemplo clássico de procedimento de gerenciamento de eventos, ocorre quando se instala um aplicativo e em seu arquivo de *logs* procura-se por erros e/ ou compatibilidade deste aplicativo com outros sistemas, com o fim de identificar mal funcionamento ou comportamento anômalo do aplicativo recém-instalado.

2.2.7 Sistema de Gerenciamento de Informações de Segurança e Eventos (SIEM)

Os sistemas de gerenciamento de informações de segurança e eventos (SIEM), conforme o definido por (KENT; SOUPPAYA, 2006) são *softwares* de registro e análise de eventos, que atuam de forma centralizada, reunindo informações geradas com base no protocolo *Syslog*. E, ainda, que os produtos de *softwares*, baseados no modelo SIEM utilizam um ou mais servidores de *logs* os quais executam as funções de análise de *logs* e um ou mais servidores de banco de dados que responsáveis pelo armazenamento dos *logs*. Os produtos SIEM, em sua maioria, suportam duas maneiras de coletar os registros de eventos dos geradores de eventos, esses geradores de eventos podem ser equipamentos ou sistemas:

AGENTLESS (sem agente): Neste modelo de coleta de registros de eventos o servidor SIEM recebe os dados dos eventos gerados individualmente pelos geradores de *logs*, não sendo necessário a existência de outros *softwares* especiais instalados nos *hosts* geradores de *logs*. Há a interação dos servidores SIEM com os *hosts* geradores de *logs*, sendo que o servidor SIEM tem a função de buscar os *logs* nos *hosts*, procedimento que é realizado via autenticação do servidor em cada um dos *hosts* envolvidos, necessitando configurações de credenciais que permitam ao servidor coletar esses *logs* em *hosts* específicos e, a ação de coleta é realizada em intervalo de tempo, previamente determinado. Há outra situação em que também, funciona sem a interação de agentes, na qual os próprios *hosts* enviam seus *logs* para o servidor SIEM, neste último procedimento a exigência que os *hosts* se autenticuem no servidor, utilizando credenciais próprias para esse fim e, então transfiram seus *logs* em intervalos pré-determinados. E, não importando se o modelo adotado seja o *host* que envia os *logs* ou o servidor quem os busque, o servidor SIEM sempre executará os procedimentos de filtragem, agregação e normalização dos eventos, além de aplicar os procedimentos configurados de análise de eventos nos registros coletados (KENT; SOUPPAYA, 2006).

AGENT-BASED (utilizando agente): Neste modelo há a instalação de um programa agente o qual é instalado nos *hosts* geradores de *logs* e, o agente terá a função de filtrar, agregar e normalizar os *logs* de acordo com as regras do servidor SIEM e, após a realização destes procedimentos o *host* gerador transmite os *logs* já tratados para o servidor SIEM, sendo que o servidor terá a função de analisar e armazenar os *logs*, normalmente essas operações ocorrem em tempo real ou o mais próximo do tempo real. Na situação em que um *host* tiver vários tipos de registros de eventos que sejam de interesse para a coleta e análise, será necessário configurar vários outros agentes, para que cada agente colete os tipos específicos de *logs*. Alguns tipos de produtos SIEM oferecem modelos de agentes já configurados em formatos genéricos, a exemplo dos formatos Syslog e SNMP (*Simple Network Management Protocol* – protocolo simples para o gerenciamento de redes). Esses agentes, em formato genérico, são utilizados principalmente para a obtenção de dados de *logs* de originadores onde exijam o formato de um agente específico e para o qual sistema não admita o uso do método de aquisição de *logs* sem o uso de agentes e, ainda, há produtos SIEM, os quais têm a capacidade de oferecer ao Administrador de rede a possibilidade de criar agentes personalizados para lidar com tipos de originadores de *logs* que não são suportados pelo produto SIEM (KENT; SOUPPAYA, 2006).

Na abordagem de (CHUVAKIN, 2016) realiza-se uma comparação entre sistemas SIEM e sistemas de gerenciamento de *logs*, focando em suas funcionalidades. Na Tabela 3 é mostrada a comparação entre sistemas SIEM e o Gerenciamento de *logs*.

Tabela 3: Comparação das funcionalidades entre sistemas SIEM e gerenciamento de *logs*

funcionalidade	sistemas SIEM	gerenciamento de <i>logs</i>
Coleta de <i>logs</i>	Coletar <i>logs</i> relevantes para a segurança	Coletar todos os <i>logs</i> , incluindo <i>logs</i> operacionais e registros personalizados de aplicativos
Armazenamento de <i>logs</i>	Manter, de forma limitada, os dados de <i>logs</i> analisados e normalizados	Manter dados brutos e dados analisados dos registros, por longos períodos
Relatórios	Relatórios focados na segurança e, relatórios em tempo real	Relatórios de uso geral e, relatórios históricos dos eventos
Análise	Correlação, classificação da ameaça e, priorização de eventos.	Análise completa de textos e, gerar marcação (<i>tags</i>)
Alerta e notificação	Relatórios avançados com foco na segurança	Alerta simples em todos os <i>logs</i>
Outras características	Gestão de incidentes e, outras análises de dados de segurança	Alta escalabilidade para coleta e pesquisa

Fonte: Adaptado de (CHUVAKIN, 2016)

2.3 A Infraestrutura para a Retenção e o Tratamento de *Logs*

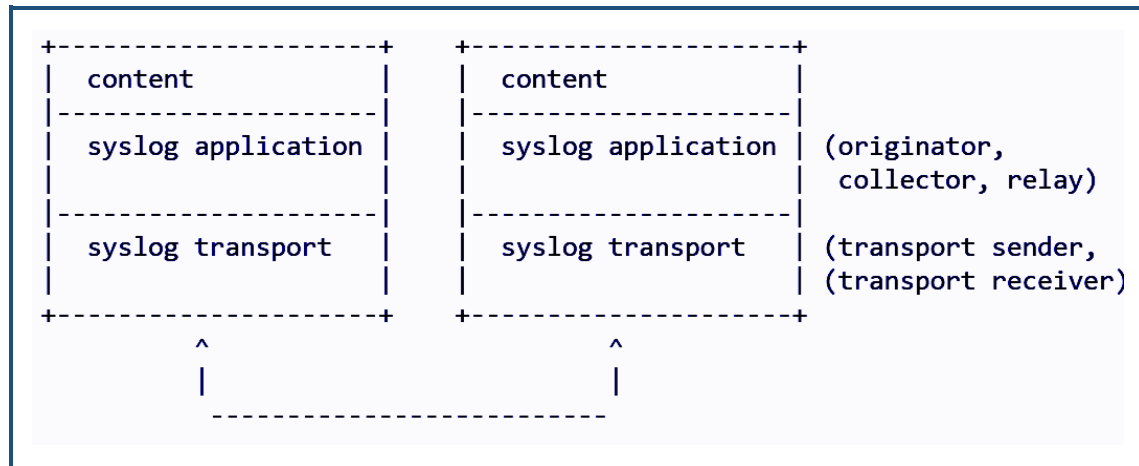
Os registros de eventos possuem enorme valor para o Administrador de Redes e Sistemas, uma vez que são o ferramental necessário para a gerência e operação de uma rede, de forma proativa, possibilitando a análise dos *logs*, para a identificação de falhas de sistemas e brechas de segurança, além de permitir a auditoria de procedimentos e, também, a aplicação de perícia forense. E, o volume de informações adquiridas torna necessário a adoção de medidas que tornem a coleta de *logs* confiável e eficaz quando da necessidade de aplicar sobre esse volume de registro uma ação de inspeção, necessitando que os registros de *logs* garantam os princípios da confiabilidade e da integridade. Em um ambiente diverso, onde coabitam diferentes sistemas operacionais e equipamentos de rede, há que se garantir a coleta e o arquivamento seguro dos *logs* das diferentes fontes geradoras de eventos, o que impõe a implementação de uma infraestrutura dedicada para a coleta e arquivamento dos eventos e, então tem-se o papel de um centralizador de *logs* (TSUNODA; KEENI, 2014).

2.3.1 O Protocolo Syslog

As definições do protocolo Syslog são abordadas na RFC 5424 (GERHARDS, 2009) e o define como sendo um protocolo utilizado para a transmissão de mensagens de notificação de eventos, utilizando uma arquitetura em camadas, o que permite o uso de um vasto número de protocolos de transporte para a transmissão das mensagens no padrão syslog, fornece também um

formato de mensagem que possibilita o uso de extensões específicas e estruturadas de fornecedores diversos. A representação das camadas do protocolo *Syslog* está definida na RFC 5424 e, a Figura 2 demonstra a interação desse modelo de camadas entre os *hosts* gerador, coletor e encaminhador dos *logs*.

Figura 2: Representação das camadas do protocolo *Syslog*



Fonte: RFC 5424 (GERHARDS, 2009)

As três camadas do protocolo *Syslog* têm funções específicas e a descrição das atividades de cada camada está demonstrada são mostradas na Tabela 4:

Tabela 4: Camadas do protocolo *Syslog*

camada	descrição
<i>syslog content</i>	é a informação de gerenciamento contida em uma mensagem <i>syslog</i>
<i>syslog application</i>	lida com a geração, a interpretação, o roteamento e o armazenamento das mensagens <i>syslog</i>
<i>syslog transport</i>	tem a função de colocar e retirar as mensagens no canal de comunicação

Fonte: Adaptado da RFC 5424 (GERHARDS, 2009)

2.3.2 O Padrão de Mensagens do Protocolo *Syslog*

O *Syslog* é um protocolo padrão para a indústria, ele registra qualquer evento ocorrido em um sistema, o *Syslog* adota o protocolo UDP (*User Datagram Protocol* – protocolo de datagrama do usuário) como o protocolo de transferência e envia esses dados do registro de eventos de todos os sistemas e dos dispositivos de segurança para um servidor de *log* no padrão *Syslog* através da porta de destino 514, podendo esta porta ser personalizada em ambientes customizados. Uma mensagem *Syslog* consiste de um pacote UDP para a porta 514 do *host* de destino contendo uma *string* com um número decimal representando a facilidade e severidade, data e hora do evento e o conteúdo da mensagem e identificação do *host* ou sistema gerador do evento. As informações fornecidas pelo gerador de uma mensagem *syslog* incluem: a) o código da facilidade e, b) o nível

de severidade. E, um serviço *syslog* adiciona informações ao cabeçalho, das informações antes de passar a entrada para o coletor *syslog*. Esses componentes incluem um ID do processo originador, um carimbo de data/ hora e o nome do *host* ou o endereço IP do dispositivo. Para ilustrar como é montada uma mensagem do protocolo, tem-se que o formato integrado da mensagem *Syslog* inclui três partes (ZHAOJUN; YONG; WENJING, 2010):

Quadro 1: Formato da mensagem do protocolo *Syslog*

<priority>timestamp hostname tag: content.

Fonte: Adaptado de (ZHAOJUN; YONG; WENJING, 2010)

A primeira parte é o campo *priority*, representa uma operação matemática entre *facility* e *severity*, que é calculada da seguinte forma, onde a prioridade é igual a facilidade, multiplicada por 8 e somada à severidade (ZHAOJUN; YONG; WENJING, 2010).

Quadro 2: Operação de cálculo da prioridade do protocolo *Syslog*

$priority = facility * 8 + severity$

Fonte: Adaptado de (ZHAOJUN; YONG; WENJING, 2010)

No campo *facility* um código é utilizado para especificar o tipo de programa que está registrando (*logging*) a mensagem. As mensagens com diferentes facilidades podem ser tratadas de forma diferente. A lista de facilidades disponíveis está definida por (GERHARDS, 2009), de acordo com a Tabela 02. Os campos *facility* e *severity* adotam o uso de código decimal, o campo *facility* possui 24 tipos e o campo *severity* possui 8 tipos, conforme os valores mostrados para o campo *facility*, na Tabela 5.

Tabela 5: Mensagens de facilidade do protocolo *Syslog*

código	palavra-chave	descrição
0	<i>kern</i>	mensagens do <i>kernel</i>
1	<i>user</i>	mensagens do nível do usuário
2	<i>mail</i>	sistema de <i>e-mail</i>
3	<i>daemon</i>	sistema de processos
4	<i>auth</i>	mensagens de segurança/ autorização
5	<i>syslog</i>	mensagens geradas internamente pelo <i>syslog</i>
6	<i>lpr</i>	subsistema de impressão
7	<i>news</i>	subsistema de mensagens da rede
8	<i>uucp</i>	subsistema UUCP
9		processo de <i>clock</i>
10	<i>authpriv</i>	mensagens de segurança/ autorização

11	<i>ftp</i>	processo FTP
12	-	subsistema NTP
13	-	registro de auditoria
14	-	registro de alerta
15	<i>cron</i>	processo de agendamento
16	<i>local0</i>	uso local 0
17	<i>local1</i>	uso local 1
18	<i>local2</i>	uso local 2
19	<i>local3</i>	uso local 3
20	<i>local4</i>	uso local 4
21	<i>local5</i>	uso local 5
22	<i>local6</i>	uso local 6
23	<i>local7</i>	uso local 7

Fonte: Adaptado de (GERHARDS, 2009)

O mapeamento entre o código *facility* e a palavra-chave não é uniforme entre sistemas operacionais e implementações de *syslog* diferentes. A segunda parte é o campo *header*, incluindo as informações de *timestamp* e *hostname*. E a terceira parte é o campo *msg*, este último composto pelas informações de *tag* e *content*, o valor da *tag* é o nome do programa ou processo que gerou a mensagem, no campo *content* estão incluídos os detalhes da mensagem (ZHAOJUN; YONG; WENJING, 2010).

O parâmetro de gravidade indica a importância de uma mensagem e leva um dos oito valores: **emergência (0), alerta (1), crítico (2), erro (3), aviso (4), notificação (5), informação (6), ou depuração (7)**. Nesses parâmetros um valor menor indica uma importância maior da mensagem e, neste contexto uma mensagem é importante quando seu parâmetro de gravidade é menor do que o valor limite (TSUNODA; *et al.*, 2009). Esse esquema é determinado pela RFC 5424, demonstrado na Tabela 6.

Tabela 6: Níveis de severidade da mensagem *Syslog*

nível	severidade	descrição
0	<i>emergency</i>	O sistema está inutilizável
1	<i>alert</i>	Uma ação deve ser tomada imediatamente
2	<i>critical</i>	Situação crítica
3	<i>error</i>	Situação de erro
4	<i>warning</i>	Situação de aviso
5	<i>notice</i>	Situação normal, mas significativa
6	<i>informational</i>	Mensagem informativa
7	<i>debug</i>	Mensagem do nível de depuração

Fonte: Adaptado de (GERHARDS, 2009)

Dependendo da implementação da solução do sistema registrador de eventos (*logger*), o protocolo *Syslog* aceita o uso de níveis para determinar o nível de registro das mensagens de *log* e, controlar a quantidade de informações a serem armazenadas em arquivos de *log*. O nível de *log* pode ser definido em tempo de execução através de um parâmetro de configuração. O modelo mostrado na Tabela 7, apresenta um modelo, apenas como exemplo, do registrador de eventos **Apache log4j**⁵, que tem sua aplicação voltada para o registro de eventos em servidores *Web* e, dele extraiu-se a informação do registro dos níveis de importância das mensagens *Syslog*. E neste cenário se um registrador de eventos usa um nível mais baixo de registro e, portanto, menor prioridade, todas as mensagens com níveis iguais ou superiores são registradas nos *logs*. Os níveis comuns, utilizados por implementações de registradores de eventos para o registro das mensagens, em ordem crescente de prioridade são os mostrados na Tabela 7 (GHOSHAL; PLALE, 2013):

Tabela 7: Níveis de registro das mensagens de *logs*

nível	descrição
ALL	é o nível de registro que admite o registro de todos os eventos de <i>logs</i>
TRACE	é o nível de registro mais detalhado, é utilizado para depurar e capturar detalhes do nível de instrução, como uma alteração para um registro particular
DEBUG	é usado para depurar um aplicativo exatamente como o nível TRACE, mas se além à declaração de linguagem de programação de alto nível, ao invés do nível de instrução de máquina.
INFO	é usado para registrar mensagens informativas que são mais úteis para monitorar e gerenciar um aplicativo durante sua execução
WARN	é usado para exceções manipuladas que podem levar à uma situação potencialmente prejudicial
ERROR	designa exceções que resultam na falha em partes do aplicativo, mas o aplicativo continuará a ser executado
FATAL	é usado para registrar eventos de erros graves que podem resultar em abortar o aplicativo
OFF	é o nível onde o registrador não captura nenhuma informação dos eventos

Fonte: Adaptado de (GHOSHAL; PLALE, 2013)

Para se entender o que ocorre em um sistema de registro de eventos, vinculado ao nível do registro das mensagens de *log*, há que se visualizar uma correlação das duas variáveis. Uma abstração dessa configuração do registrador de eventos é mostrada na Tabela 8:

⁵ <https://logging.apache.org/log4j/2.0/>

Tabela 8: Nível do evento versus nível de configuração do registrador de eventos

		Nível de configuração do registrador de eventos						
Nível do evento		TRACE	DEBUG	INFO	WARN	ERROR	FATAL	OFF
	ALL							
	TRACE							
	DEBUG							
	INFO							
	WARN							
	ERROR							
	FATAL							
	OFF							

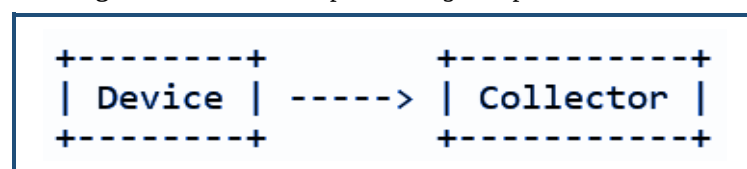
Fonte: Adaptado de (APACHE.ORG, 2017)

LEGENDA ATIVADO
 DESATIVADO

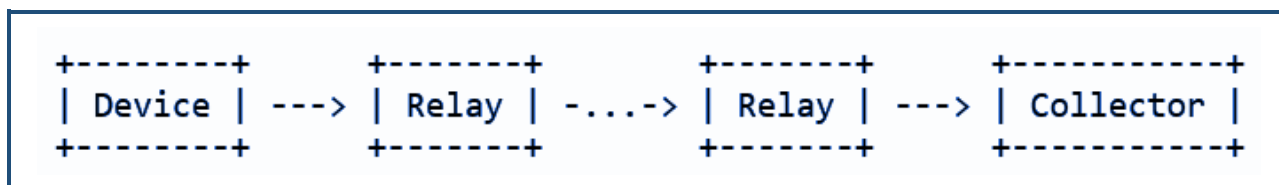
Esses níveis de configuração nos registradores de eventos, determina o volume de dados coletados e o detalhamento das mensagens dos eventos registrados, impactando diretamente no tamanho dos arquivos gerados e na qualidade das informações coletadas. São questões que influenciam a compreensão dos diferentes níveis de informação que podem ser armazenados em arquivos de *logs*; a relevância de dados dos registros de eventos e; a análise das opções e recursos, que deixam opções a serem avaliadas entre melhorar a qualidade das informações coletadas e o dispêndio de recursos na análise de dados dos *logs* (GHOSHAL; PLALE, 2013).

2.3.3 Serviço de Coleta de Logs – Modelo Loghost Centralizado

Em conformidade com a RFC 3195 (NEW; ROSE, 2001), o serviço Syslog suporta três funções de operação: a) dispositivo; b) encaminhador e c) coletor. Dispositivos e Coletores atuam como fontes e armazenadores, respectivamente, das entradas do Syslog, no caso mais simples, apenas um dispositivo e um coletor estão presentes no cenário. A relação entre os dispositivos e os coletores é, em sua essência, muitos-para-muitos. Ou seja, um dispositivo pode se comunicar com muitos coletores e da mesma forma, um coletor pode se comunicar com muitos dispositivos. Um encaminhador opera em ambos os modos, aceitando as entradas do Syslog provenientes dos dispositivos e de outros encaminhadores, então encaminham essas entradas para os coletores ou outros encaminhadores.

Figura 3: Modelo de captura de logs: dispositivo – coletor

Fonte: RFC 3195 (NEW; ROSE, 2001)

Figura 4: Modelo de captura de *logs*: dispositivo – encaminhador – coletor

Fonte: RFC 3195 (NEW; ROSE, 2001)

A rápida detecção de anomalias em uma rede, exige que se tenha o acesso consolidado ao volume de informações coletadas via registro de eventos e, a adoção do recurso de centralizadores de *logs* permite ao Administrador monitorar a rede e identificar os problemas e possibilita, ainda, a rápida adoção de uma solução (MICROSOFT, 2017)⁶.

2.3.4 Rotacionamento de Logs

A função do protocolo *Syslog*, denominada *Logrotate* existe na maioria dos sistemas operacionais e, normalmente, um *log* capturado pelo *Syslog* é substituído dentro de um período de tempo especificado e, portanto, não se consegue, após o período de rotação, fazer referência ao registro antigo. O registro mais antigo disponível é de apenas quatro semanas, porque a configuração padrão do *Logrotate* gira o *log* uma vez por semana e verifica os registros antigos. Este problema pode ser resolvido até certo ponto, aumentando o número de vezes do rotacionamento. E, outra forma, com maior utilidade é copiar os registros dos *logs* mais antigos para um dispositivo de armazenamento de grande capacidade, por exemplo para um *storage*⁷ na rede (TOMONO; *et al.*, 2009).

No caso de máquinas com sistema operacional *Windows*, um *log* é transferido pelo serviço nativo *NTsyslog* e, no caso de máquinas com sistema operacional *Linux*, os *logs* são transferidos pelo serviço *Syslog-ng*. Em seguida, os padrões são unificados de acordo com o formato do protocolo *Syslog* e transferidos para um servidor de coleta de *logs*, após cada um ter sido transferido para um servidor com o mesmo sistema operacional. O servidor para a coleta de *logs* rotaciona todos os *logs* utilizando a função *Logrotate*. Os diretórios que descrevem os tipos de *logs* são criados, de antemão, no sistema de armazenamento de alta capacidade (TOMONO; *et al.*, 2009).

Dependendo da necessidade de coleta dos dados de eventos, considerando a quantidade de dados e outros requisitos, como tamanho do armazenamento, ou a época em que os eventos aconteceram, estes requisitos podem ser utilizados para determinar com qual frequência os eventos

⁶ Coletar Eventos: [https://technet.microsoft.com/pt-br/library/cc748890\(v=ws.11\).aspx](https://technet.microsoft.com/pt-br/library/cc748890(v=ws.11).aspx)

⁷ *Storages* são dispositivos projetados especificamente para o armazenamento de grandes volumes de dados, tipicamente, utilizam conexões de alta velocidade em uma estrutura de rede.

de sistema devem ser coletados e arquivados. As três configurações base de frequência, de armazenamento e renovação do arquivo de *log* – o rotacionamento – suportado pelo padrão do protocolo *Syslog* é determinado por uma variável chamada `FREQUÊNCIA`, a qual admite três valores básicos implementados no protocolo, os quais são: `DIÁRIO`, `SEMANAL` e `MENSAL` (CARDO, 2011).

E, analisando a variável `FREQUÊNCIA` no valor `DIÁRIO`, o arquivo de *log* do dia anterior será processado, com prazo de 24 horas e, não há limitações no dia da semana, mês ou ano, as últimas 24 horas anteriores serão processadas em um arquivo único (CARDO, 2011).

E se a variável `FREQUÊNCIA` for definida para o valor `SEMANAL`, os arquivos de *log* dos últimos sete dias serão processados. No padrão do protocolo *Syslog*, uma semana é definida como sendo de domingo a sábado e, os arquivos de *log* serão sempre processados em um período de domingo a sábado, totalizando sete dias e, o intervalo de datas para o valor `SEMANAL` pode ultrapassar os limites de mês e ano, não importando a data de início ou fim da semana (CARDO, 2011).

Na situação da escolha do registro da variável `FREQUÊNCIA` no valor `MENSAL`, limita o processamento do arquivo de *log* para um mês do calendário físico, os arquivos de eventos do mês do calendário anterior serão processados e, o intervalo de datas é desde o primeiro dia do mês até o último dia do mês, os limites dos meses nunca são cruzados, o que se solicitou para registro dentro de um mês, não invade os registros dos dias do próximos mês, mesmo que o registro `MENSAL` tenha iniciado, parcialmente, depois da data de início do mês, esse registro se finda ao dia que encerra aquele mês específico (CARDO, 2011).

3

TRABALHOS RELACIONADOS

Neste capítulo serão apresentados alguns trabalhos e abordagens relacionados ao tratamento dos registros de eventos. Os trabalhos aqui elencados tratam sobre os temas da auditoria, da análise, da mineração de *logs* e sobre o Sistema de Gerenciamento de Informações de Segurança e Eventos (SIEM – *Security Information and Event Management*). A proposta deste capítulo é demonstrar que a abordagem da pesquisa, no referente à análise de *logs*, propõe um método próprio, utilizando *Scripts Windows PowerShell* para localizar eventos relevantes ao Gerenciamento da Segurança em uma rede local.

3.1 Análise, Mineração e Auditoria de *Logs*

A abordagem de visualização para a análise de *logs*, é discutida por [\(STEARLEY, 2004\)](#), afirmando que esta é uma abordagem popular para análise de *logs* e, que embora a apresentação de visualização de *logs* em gráficos bidimensionais ou tridimensionais sejam representações muito interessantes de dados de *logs*, nenhuma métrica de similaridade ou tipo de apresentação é amplamente aceita como uma resposta efetiva ao problema de análise de *logs*.

O estudo dos eventos registrados em arquivos de *logs* de sistema com o objetivo de se caracterizar o comportamento de um sistema e de usuário, conforme o abordado na pesquisa de [\(LIM; SINGH; YAJNIK, 2008\)](#), tem sido um foco de diversas pesquisas, como exemplo os *logs* de navegação *Web* onde contém padrões de pesquisa dos usuários e o comportamento da navegação, sendo que estes registros foram estudados para melhorar o uso de *sites web* e, também, para fornecer de forma direcionada aos usuários publicidade de produtos que tenham relação com suas pesquisas e comportamentos. Os registros do protocolo *Syslog*, *logs* de transações e registros de erros também foram matéria de mineração de dados, com o fim de detectar falhas ou comportamento anômalo em sistemas computacionais. E, outra técnica de mineração é a visualização do registro de eventos, que se tornou uma área fértil para a pesquisa de mineração de dados, onde se emprega a técnica de visualização abstrata do arquivo de *logs*, como ferramenta para determinar o estado de um sistema.

Para (SONG; LUO; CHEN, 2008) no que se refere à mineração de padrões de comportamento a mineração de fluxo de trabalho têm alguns pontos comuns, como ambos usam padrões de fluxo de trabalho como símbolos para apresentar os resultados minerados e, algumas abordagens de mineração podem ser aplicadas em ambos os dois casos.

Em pesquisa realizada por (VAARANDI; PIHEL GAS, 2014), afirma-se que nos últimos anos, o estabelecimento de métricas adequadas para se medir a segurança de sistemas recebeu uma atenção cada vez maior. E, considera que os *logs* de segurança contêm grandes quantidades de informações que são essenciais para se criar muitas métricas de segurança. E que infelizmente, os registros de segurança são conhecidos por gerarem arquivos enormes, o que torna análise destes grandes volumes de dados, uma tarefa difícil. Afirma, também, que as pesquisas recentes sobre métricas de segurança se concentraram em conceitos genéricos e que a questão da coleta de métricas de segurança com métodos de análise de *log* não foi bem estudada. A pesquisa se concentrou no uso de técnicas de análise de *logs* com o fim de se coletar métricas de segurança a partir de registros de segurança de tipos comuns a exemplo de *logs* de alarme de sistemas, de detecção de intrusão de rede, *logs* de estações de trabalho e, conjuntos de dados *Netflow*. Essa pesquisa também descreve um *framework* de produção com o fim de coletar e relatar técnicas para métricas de segurança, o qual se baseia em novas tecnologias de código aberto para enorme volume de dados.

Na proposta de mineração de *logs* por (SONG; *et al.*, 2017) temos que a mineração de processos está atraindo enorme interesse devido a necessidade de se extrair, de forma automática, procedimentos de inteligência de negócios (BI – *Business Intelligence*) que são iniciados a partir de *logs* de eventos. Há inúmeras técnicas de mineração de processos, como: a) verificação de conformidade, b) aprimoramento (ou extensão) de modelo de processos e c) avaliação de desempenho, que para serem aplicadas necessitam uma base de *logs* de eventos e de modelos de processo para a tomada de decisões.

Outra proposta para a utilização das técnicas de mineração de dados em arquivos de *logs* é abordada por (ZHUGE; VAARANDI, 2017), pesquisa na qual apresenta-se que os métodos de mineração de dados são uma alternativa factível para o fim de efetuar análises nos arquivos de *logs* e, como técnica de mineração, o uso dos algoritmos de agrupamento de dados é uma das abordagens mais comumente propostas nas pesquisas acadêmicas. Em razão dos arquivos de *logs* de eventos serem, em sua forma mais básica, um amontoado de elementos textuais, onde cada evento é descrito por uma única linha de *log* de eventos, os algoritmos de agrupamento (*cluster*) de dados foram projetados para a mineração de padrões de linha destes arquivos de *logs*, nestes padrões de linha são detectados, por exemplo, informações de fornecimento de senha incorreta em

procedimentos de autenticação de credenciais e, fornecem informações valiosas sobre os tipos de eventos que ocorrem comumente e, ainda, estas informações podem ser utilizadas para vários fins, por exemplo, o desenvolvimento de regras para o monitoramento de *logs* e, também para a correlação de eventos, uma vez que os algoritmos de agrupamento de dados também permitem a identificação de pontos de dados anormais, então estes algoritmos são úteis para destacar eventos incomuns.

3.2 Gerenciamento e Correlação de Eventos de Segurança

No trabalho de pesquisa de (MYERS; GRIMAILA; MILLS, 2011), defende-se que o procedimento de análise de dados em arquivos de registro de eventos apresenta benefícios ao processo de gerenciamento de eventos segurança e aos recursos computacionais das organizações. Enfatiza a extrema utilidade da aplicação da análise de *logs* e a possibilidade de detecção de eventos de segurança a partir dos dados de arquivos de *logs*. E, o foco da pesquisa está nos benefícios proporcionados pela abordagem da correlação de eventos de segurança (SIEM), dando a essa abordagem – correlação de eventos de segurança – um posicionamento de ser subconjunto da atividade de análise de *logs*, ressaltando também que a análise de *logs* de forma distribuída, que é descentralizando os *logs* principais em estações coletoras distribuídas, possibilita a análise de um conjunto menor de eventos por estação coletora.

Outra pesquisa acerca do gerenciamento e correlação de eventos de segurança (SIEM) é conduzida por (CHENG; *et al.*, 2013), onde a abordagem é direcionada para a aplicação do algoritmo de análise de agrupamento *k-means*⁸, objetivando estabelecer uma análise de segurança de alto desempenho e para enfrentar o desafio de normalizar, centralizar e correlacionar a análise de informações de eventos em tempo real, com o objetivo de facilitar a identificação de estado de execução atual no ambiente de destino e, nesta pesquisa é proposto um modelo de plataforma para o gerenciamento de informações de segurança e eventos (SIEM), afirmando-se tratar de uma plataforma de próxima geração. E como modelo de execução é estabelecido um laboratório de análise de segurança, denominado SAL (*Security Analytics Lab*) que foi projetado e implementado com base na técnica de gerenciamento de dados em memória (*In-Memory Data Management*), técnica esta que permite organizar, acessar e processar diferentes tipos de informações de eventos através de armazenamento e interface central consistentes. Nessa pesquisa é apresentada a arquitetura multicore no módulo de correlação de eventos para a plataforma do laboratório de análise de segurança, arquitetura pela qual é possível a execução, em paralelo, das tarefas de

⁸ *k-means* (James MacQueen, 1967) é um método de quantização vetorial que foi desenvolvido, originalmente, no processamento de sinais e que se tornou de uso geral para a análise de *cluster* no processo de mineração de dados.

correlação de eventos, utilizando diferentes recursos de computação e, o algoritmo *k-means* é implementado como um exemplo de possíveis agrupamentos de eventos e algoritmos de correlação e, no laboratório proposto há diversos experimentos que são conduzidos e analisados com o fim de demonstrar que o desempenho da análise pode ser significativamente melhorado através da aplicação da arquitetura multi-núcleo no procedimento de correlação de eventos.

3.3 Análise de *Logs* de Autenticação em Rede

A análise de *logs* de autenticação é um campo de estudo que tem relevância para a pesquisa do Gerenciamento de Segurança e, uma das aplicações é o Gerenciamento da Segurança em redes internas. Na pesquisa de (TSUNODA; *et al.*, 2009), constata-se que uma das aplicações mais importantes para a análise de *logs* é o gerenciamento de segurança para a intranet, pois várias ameaças à segurança surgem a cada dia e, um Administrador de Segurança deve investigar as informações de registro geradas por todos os *hosts*, incluindo-se nesse conjunto todos os equipamentos servidores de rede e as estações de usuários da rede local. Como exemplo cita as informações coletadas de *logs* que foram gerados por servidores *Web*, uma vez que estes serviços são alvos constantes de tentativas de ataques e, que a forma destas tentativas são informações valiosas aos Administradores de Segurança.

Uma das diversas observações que podem ser realizadas na análise de *logs* de rede local, conforme (TSUNODA; *et al.*, 2009) é a visualização de tentativas de usuários em instalar algum *software* não autorizado em suas estações e, as informações sobre essas ocorrências podem ser recuperadas a partir da análise de *logs* do sistema do *host* cliente, mas na imensa maioria dos cenários do Gerenciamento da Segurança de redes, os dados de registros de eventos se perdem no enorme volume de dados que não deixam de crescer diariamente, se misturando a diversas outras informações mais comuns e menos agressivas. E para se descobrir as informações dos *logs* de eventos, as quais sejam relevantes para a gestão da segurança, separando informações meramente operacionais das informações críticas à segurança não é tarefa fácil. Há situações em que as informações nos *logs* são geradas por eventos operacionais, mas que podem vir a caracterizar uma evidência de tentativa de acesso indevido ou mesmo identifica uma varredura de vulnerabilidades do sistema a ser atacado, então como propósito da Administração de Segurança, deve-se coletar todas as informações dos registros de eventos, independente das causas que geraram a informação no registro de eventos.

Na análise da pesquisa de (TOMONO; *et al.*, 2009) foca na implementação de um sistema interno de auditoria baseado nos dados recolhidos de um sistema de registro de eventos e, evidencia que os registros de auditoria são muito importantes para os procedimentos de controle

interno, reforçando que embora qualquer controle interno que não tenha a uma função dedica à auditoria seja um sistema de controle inútil. E destaca, ainda, que grandes empresas, como a IBM e a *Sun Microsystems*, lançaram seus sistemas de controle interno, para suprir a enorme necessidade de análise dos registros de eventos em grandes corporações, mas de outro lado atenta para o fato de que os sistemas terceirizados com o objetivo de proceder o controle interno acarretam em um custo maior para as organizações. Custo este que é proporcional ao tamanho da organização e, reforça que a premissa do processo de controle interno é a ação de gerenciamento e monitoramento de cada serviço com base em padrões e procedimentos prescritos, que têm o objetivo de fornecer a garantia da operação eficiente, efetiva e legal das organizações e que a ação de gerenciamento e monitoramento deve ser executada de formar a evitar-se fraudes e erros na obtenção dos dados para o procedimento de auditoria.

Reforçando que um sistema de controle interno precisa incluir os seguintes elementos: a) autenticação; b) autorização e, c) auditoria:

- **Autenticação:** é um processo no qual as credenciais fornecidas por um usuário são comparadas às credenciais que estão no banco de dados de informações de usuários autorizados em um sistema operacional de uma estação ou em um servidor de autenticação;
- **Autorização:** é a função de especificar direitos ou privilégios de acesso aos recursos relacionados à segurança da informação e à segurança de um sistema computacional em geral e ao controle de acesso em particular e, o termo “autorizar” significa definir uma política de acesso;
- **Auditoria:** é um exame sistemático com o fim de analisar os procedimentos e registros de um sistema, a fim de garantir o atendimento às regras pré-estabelecidas ou às legislações próprias sobre o tema em análise.

Os *logs* incluem informações sobre autenticação e autorização. Em outras palavras, a auditoria dos registros é o foco fundamental em um sistema de Controle Interno.

3.4 Comparação entre as Propostas de Análise de Logs

As propostas estudadas foram classificadas em cinco aspectos: a) técnica utilizada; b) a abordagem da técnica; c) o que a pesquisa entrega como resultado; d) o modelo de apresentação da solução, linha de comando ou ambiente gráfico e; e) o ferramental disponibilizado para a implementação da solução. E na Tabela 9 está apresentado o comparativo entre as propostas.

Tabela 9: Comparação das propostas de métodos de análise de *logs*

PROPOSTA	TÉCNICA	ABORDAGEM	ENTREGA	GUI	FERRAMENTAL
STEARLEY, 2004	COMPARA FERRAMENTAS	ANÁLISE DE LOGS	PROPOSTA DE ANÁLISE COMPARATIVA ENTRE FERRAMENTAS	NÃO	SCRIPT/ ALGORITMO
LIM; SINGH; YAJNIK, 2008	COMPARA FERRAMENTAS	ANÁLISE DE LOGS	PROPOSTA DE ANÁLISE VISUAL DE EVENTOS	SIM	SCRIPT
SONG; LUO; CHEN, 2008	COMPARA FERRAMENTAS	ANÁLISE DE COMPORTAMENTO/ MINERAÇÃO DE LOGS	PROPOSTA DE FLUXO DE MINERAÇÃO DE LOGS	NÃO	ALGORITMO
TSUNODA; <i>et al.</i> , 2009	COMPARA MÉTODOS	ANÁLISE DE TRÁFEGO DE LOGS	PROPOSTA DE MECANISMO DE AQUISIÇÃO DE LOGS	NÃO	ALGORITMO
TOMONO; <i>et al.</i> , 2009	COMPARA TÉCNICAS E FERRAMENTAS	ANÁLISE E GERENCIAMENTO DE LOGS	PROPOSTA DE MÉTODO PARA O ARMAZENAMENTO DE LOGS	NÃO	SCRIPT
MYERS; GRIMAILA; MILLS, 2011	COMPARA FERRAMENTAS	ANÁLISE DE LOGS	PROPOSTA DE CORRELAÇÃO DE EVENTOS DISTRIBUÍDOS (SIEM)	NÃO	SCRIPT/ ALGORITMO
CHENG; <i>et al.</i> , 2013	COMPARA MÉTODOS	ANÁLISE DE INFORMAÇÕES DE EVENTOS	PROPOSTA DE PLATAFORMA PARA GERENCIAMENTO DE INFORMAÇÕES DE SEGURANÇA E EVENTOS (SIEM)	NÃO	ALGORITMO
VAARANDI; PIHELGAS, 2014	COMPARA MÉTODOS E FERRAMENTAS	ANÁLISE DE LOGS	PROPOSTA DE MODELO PARA COLETA DE MÉTRICAS DE SEGURANÇA EM LOGS	SIM	SCRIPT
SONG; <i>et al.</i> , 2017	COMPARA MÉTODOS	ANÁLISE DE LOGS/ COMPARAÇÃO DE PROCESSOS	PROPOSTA PARA ALINHAR LOGS DE EVENTOS COM MODELOS DE PROCESSO	NÃO	ALGORITMO
ZHUGE; VAARANDI, 2017	COMPARA FERRAMENTAS	ANÁLISE DE LOGS	PROPOSTA DE MELHORIA NA PERFORMANCE DE ALGORITMOS	NÃO	ALGORITMO
NOSSA PROPOSTA	COMPARA TÉCNICAS E FERRAMENTAS	ANÁLISE E GERENCIAMENTO DE LOGS	PROPOSTA DE FERRAMENTAL PARA A ANÁLISE E GERENCIAMENTO DE LOGS	SIM	SCRIPT

Fonte: Próprio autor

O destaque na nossa proposta é a abordagem no fornecimento de relatórios após o procedimento de análise dos *logs*, o que permite ao Administrador de Rede visualizar as informações de interesse, armazenar os relatórios conforme suas necessidades e, ainda, possibilita o uso do ferramental resultante (*scripts* PS1) em diferentes plataformas *Windows*, pois há a completa compatibilidade de plataforma e de definição dos caminhos dos arquivos em análise.

4

MODELO DE ANÁLISE DE LOGS PARA AUDITORIA DE REGISTRO DE EVENTOS

Após a definição do referencial teórico sobre o tema abordado, buscou-se implementar uma infraestrutura de coleta de *logs*, com base no modelo de *loghost* centralizado, atribuindo-se a função de coleta de *logs* a uma estação com sistema operacional *Windows 7 Professional SP1* e a função de servidor originador de *logs* será de um sistema operacional *Windows Server 2012*, rodando o *Active Directory*, de onde serão coletados os *logs* e, então, direcionados à estação de coleta. Ao final do capítulo é proposto um modelo para a análise generalista de *logs* de eventos, com foco e abordagem no Gerenciamento e na Auditoria de *logs* de eventos.

4.1 Cenário para o Estudo de Caso

No Instituto Federal do Amapá (IFAP) a estrutura de Rede e Serviços de Rede está distribuída entre 06 servidores físicos e 12 serviços virtualizados, onde são executados os serviços: DHCP; *Proxy*, *Firewall*; Antivírus; Autenticação de Usuários; Servidor de Arquivos; VoIP; RNP CAFE; *Web Server*; Armazenamento de Dados – *storage*; DNS; Gerenciamento de Projetos; Gerenciamento de *Tickets* de Serviços de TI; Sistemas Acadêmicos; Sistemas Administrativos, dentre outros serviços.

Analisando a atual estrutura de rede e serviços da Diretoria de TI do IFAP, constatou-se que não há um serviço de coleta, armazenamento e tratamento dos registros de eventos, então tomou-se a decisão, em consonância com a Direção de TI/ IFAP, de implementar um novo serviço na rede local, que é o serviço de coleta centralizada de *logs* e, esta tarefa é a base deste trabalho de pesquisa, tomando-se como universo de análise, os dados reais do histórico de eventos de sistema registrados no servidor de autenticação do domínio `ifap.local`. Algumas configurações e adaptações dos sistemas envolvidos foram realizadas e, essas adaptações estão detalhadas mais adiante neste trabalho.

4.1.1 Configuração do Servidor Origem dos Logs – Windows Server 2012 (AD)

Os *hosts* que utilizam o sistema operacional *Windows* usam o serviço *NTSyslog* para enviar mensagens do protocolo *Syslog* e, para a configuração do ambiente de visualização de *logs* de eventos do *NTSyslog* com o intuito de enviar as informações dos registros de eventos de um *host* coletor, o número de mensagens do protocolo *Syslog* e o tempo de transmissão das mensagens não são controláveis. Essas características são completamente dependentes do comportamento dos sistemas operacionais, programas de aplicativos e da implementação do *Syslog*. Então, o tempo de sincronia e o volume das mensagens enviadas do dispositivo gerador para o dispositivo coletor, fica a cargo de inúmeras variáveis e do comportamento dos sistemas operacionais *Windows* envolvidos (TSUNODA; *et al.*, 2009).

A funcionalidade de encaminhamento de *logs* de eventos foi introduzida no sistema operacional *Windows Server*, a partir da versão 2008, função que permite ao Administrador centralizar os *logs* de eventos, tanto de servidores, quanto de estações clientes, o que facilita o monitoramento dos eventos, evitando a necessidade de se conectar a cada sistema independente que gera seus próprios *logs*. A função de redirecionamento utiliza o padrão DTMF (*Distributed Management Task Force* – Grupo de Trabalho de Gerenciamento Distribuído), sob o recurso denominado *WS-Eventing* (*Web Services Eventing* – Eventos baseados em Serviços da Web), o qual faz parte do protocolo aberto *WS-Man* (*Web Services Management* – Gerenciamento de Serviços Web) integrado ao sistema operacional *Windows Server* como parte do modelo WMF (*Windows Management Framework* – Modelo de Gerenciamento do Windows) (SMITH, 2015).

Na implementação da proposta demonstra-se os procedimentos para configurar o encaminhamento de *logs* de eventos no sistema operacional *Windows Server* 2012, configurando um servidor de origem e uma estação *Windows 7 Professional* SP1, que atuará como coletor, registrando-se que para que uma estação *Windows*, atue como *loghost* centralizado, necessita ser superior à versão *Windows Vista* e, para atuar como máquina originadora de *logs*, deve ser posterior ao *Windows Server* 2003 SP1 e *Windows XP* SP2 (SMITH, 2015).

Não há a necessidade de se instalar um agente de coleta, já que o encaminhamento de *logs* de eventos usa as tecnologias integradas (*built-in*) ao *Windows Server* 2012. Um coletor pode encaminhar registros de eventos, para outro coletor e processar muitos eventos por segundo, tornando o encaminhamento muito escalável e, pode utilizar HTTPS (*Hyper Text Transfer Protocol Secure* – Protocolo de Transferência de Hipertexto Seguro), para o transporte de mensagens seguras (SMITH, 2015).

Os procedimentos realizados para a configuração do encaminhamento de *logs*, no Servidor *Windows Server 2012*, foram as seguintes:

a) O encaminhador dos *logs* de eventos

Um ou mais dispositivos podem ser configurados como coletores de *logs* de eventos e, nestes dispositivos, configura-se as assinaturas que buscam os *logs* desejados de quaisquer computadores origem e, não há a necessidade de se proceder à nenhuma configuração especial nos computadores originadores de *logs*, exceto a configuração dos serviços de gerenciamento remoto do *Windows* (WinRM – *Windows Remote Management*) que deve ser habilitado e, também, devem ser habilitadas as exceções do serviço WinRM no *Windows Firewall* e, ainda, a conta do computador do coletor deve ter permissão de leitura nos *logs* aos quais se irá coletar no originador.

Existem variações nesta configuração padrão, que não serão discutidas neste trabalho e, há especificidades próprias de cada ambiente que se pretenda aplicar este mesmo cenário, as quais não serão objeto de análise nesta pesquisa como algumas configurações especiais, a exemplo: a) configurar notificações *push* e b) configurar uma conta de usuário para autenticar nos computadores de origem, mas ressaltando que o ambiente de testes deste trabalho é o cenário real do IFAP com a estrutura de domínio *Active Directory* (AD).

b) Configurar o computador origem dos eventos – *Windows Server 2012*

Para a ativação do encaminhamento de *logs* de eventos em um controlador de domínio no sistema operacional *Windows Server 2012* e, na abordagem desta pesquisa, o servidor: SVR-AD01, necessita habilitar-se ao serviço do gerenciamento remoto do *Windows* (WinRM). E, para a facilidade no acompanhamento das configurações, será adotado o uso do ambiente *Windows PowerShell*, que é uma interface de linha de comando baseada em tarefas e, em uma linguagem de *scripts* cuja funcionalidade está voltada para a administração de sistemas *Windows*.

Então, para a inicialização do ambiente de testes deve-se habilitar o serviço WinRM no controlador de domínio de origem dos eventos (*Domain Controller* – Controlador de Domínio), todos os comandos abaixo exigem que se tenha o privilégio de Administrador da máquina local onde roda o serviço do AD.

No Quadro 3 é apresentada a instrução, executada no ambiente de linha de comando na máquina SVR-AD01, dentro do ambiente *Windows PowerShell*:

Quadro 3: Verificação do estado do serviço WinRM no sistema *Windows Server 2012*

```

Windows PowerShell
PS C:\Users\Administrador>winrm get winrm/config
Config
    MaxEnvelopeSizekb = 500
    MaxTimeoutms = 60000
    MaxBatchItems = 32000
    MaxProviderRequests = 4294967295
    Client
        NetworkDelayms = 5000
        URLPrefix = wsman
        AllowUnencrypted = false
        Auth
            Basic = true
            Digest = true
            Kerberos = true
            Negotiate = true
            Certificate = true
            CredSSP = false
        DefaultPorts
            HTTP = 5985
            HTTPS = 5986
        TrustedHosts
    Service
        RootSDDL =
O:NSG:BAD:P(A;;GA;;;BA)(A;;GR;;;IU)S:P(AU;FA;GA;;;WD)(AU;SA;GXGW;;;WD)
        MaxConcurrentOperations = 4294967295
        MaxConcurrentOperationsPerUser = 1500
        EnumerationTimeoutms = 240000
        MaxConnections = 300
        MaxPacketRetrievalTimeSeconds = 120
        AllowUnencrypted = false
        Auth
            Basic = false
            Kerberos = true
            Negotiate = true
            Certificate = false
            CredSSP = false
            CbtHardeningLevel = Relaxed
        DefaultPorts
            HTTP = 5985
            HTTPS = 5986
        IPv4Filter = *
        IPv6Filter = *
        EnableCompatibilityHttpListener = false
        EnableCompatibilityHttpsListener = false
        CertificateThumbprint
        AllowRemoteAccess = true
    Winrs
        AllowRemoteShellAccess = true
        IdleTimeout = 7200000
        MaxConcurrentUsers = 2147483647
        MaxShellRunTime = 2147483647
        MaxProcessesPerShell = 2147483647
        MaxMemoryPerShellMB = 2147483647
        MaxShellsPerUser = 2147483647

```

Fonte: Cenário real do IFAP – próprio autor

Caso a tela acima não seja a resposta que se obtenha, o serviço WinRM não estará habilitado, configure-o executando o comando: **c:\winrm quickconfig**

Então, após a ativação do serviço WinRM, necessita-se configurar a segurança para o acesso aos *logs*, pois antes que a estação *Windows 7 Professional SP1* possa atuar como coletora dos *logs* de eventos, é necessário adicionar a conta de computador do coletor ao grupo de Leitores

de *log* de eventos, no servidor SVR-AD01, registrando que o nome de máquina da estação coletora neste cenário é: RTDTIPC012725\$. Para adicionar a conta da máquina coletora utiliza-se a linha de comando do Quadro 4, no ambiente *Windows PowerShell*:

Quadro 4: Adicionar máquina coletora ao grupo de leitores de eventos no servidor

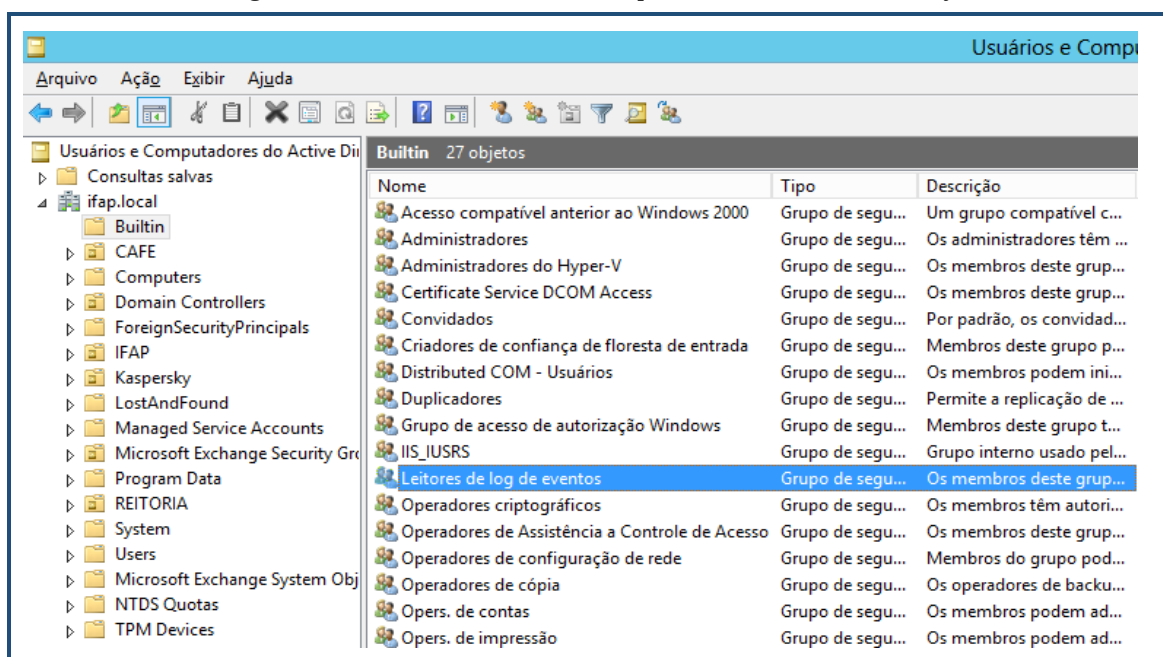
```
c:\Add-ADGroupMember -identity 'Leitores de log de eventos' -member RTDTIPC012725
```

Fonte: Cenário real do IFAP – próprio autor

- Confirmar a adição da estação coletora ao **grupo Leitores de log de eventos**, no *Windows Server 2012*, utiliza-se a **ferramenta administrativa Usuários e Computadores do Domínio**, abrindo-se no painel da esquerda, abaixo da árvore do Domínio (ifap.local) a **Unidade Organizacional Builtin**;

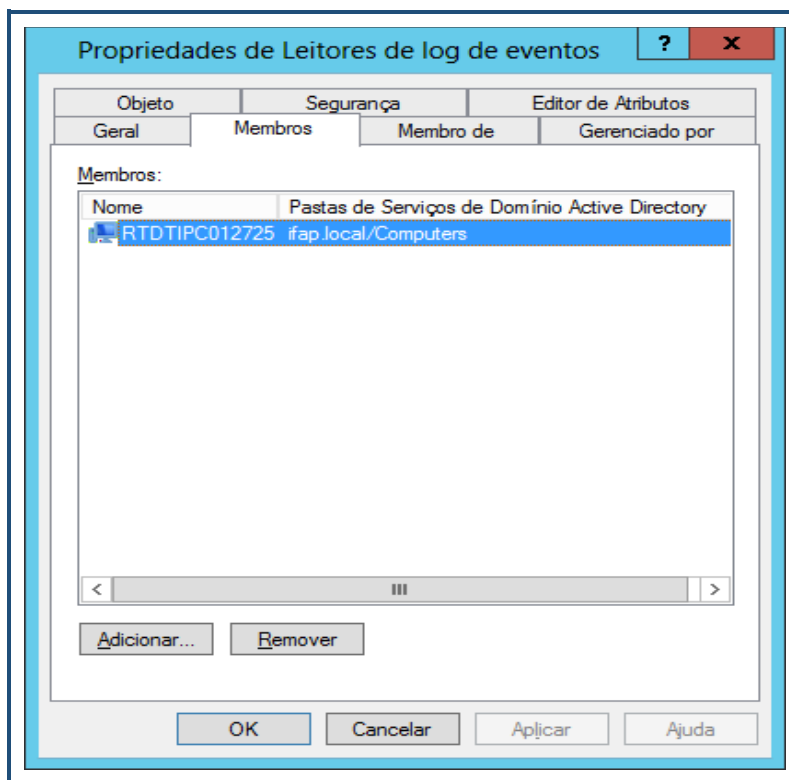
A Figura 6 apresenta a janela *Usuários e Computadores do Domínio*, onde se pode confirmar que a Unidade Organizacional *Builtin* apresenta, no lado da esquerda, o objeto do Grupo de segurança *Leitores de logs e eventos*:

Figura 6: Ferramenta *Usuários e Computadores do Active Directory*



Fonte: Cenário real do IFAP – próprio autor

- Na lista à esquerda, abrindo as **Propriedades** do grupo **Leitores de log de eventos**, abre-se a **Guia Membros**, confirma-se que a máquina RTDTIPC012725\$ foi inserida com sucesso como membro do grupo.

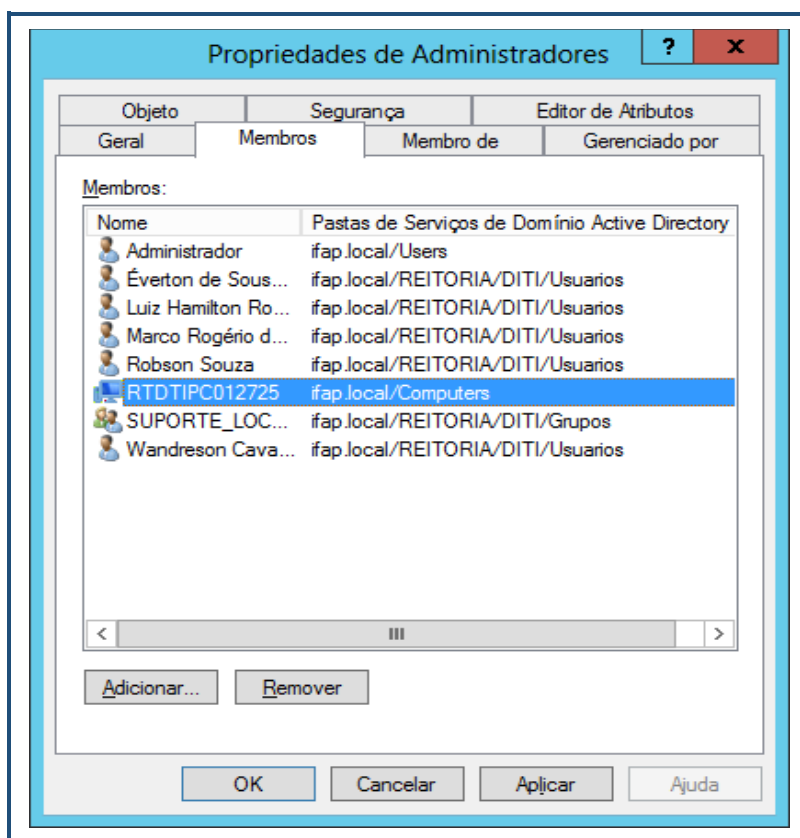
Figura 7: Confirmação da adição da máquina coletora de eventos

Fonte: Cenário real do IFAP – próprio autor

Para que exista uma relação de confiança entre o Servidor de origem dos *logs* e a estação que atuará como coletor, há a necessidade de adicionar a estação coletora de eventos ao grupo de Administradores no Servidor de origem dos *logs*.

- No Servidor de origem dos *logs*, abrir o módulo das ferramentas administrativas e, escolher o módulo de **Usuários e Computadores do Domínio**, clicar na **Unidade Organizacional Built-in** e, abrir as **Propriedades** o grupo **Administradores**, adicionando-se o nome da estação coletora, no cenário de testes é a estação **RTDTIPC012725\$** como membro do grupo, ressaltando a necessidade de se adicionar o símbolo **\$** ao final do nome da estação, indicando que se trata de uma conta de máquina e não de usuário ou grupo de usuários;

Figura 8: Adicionar a estação coletora de eventos ao grupo Administradores

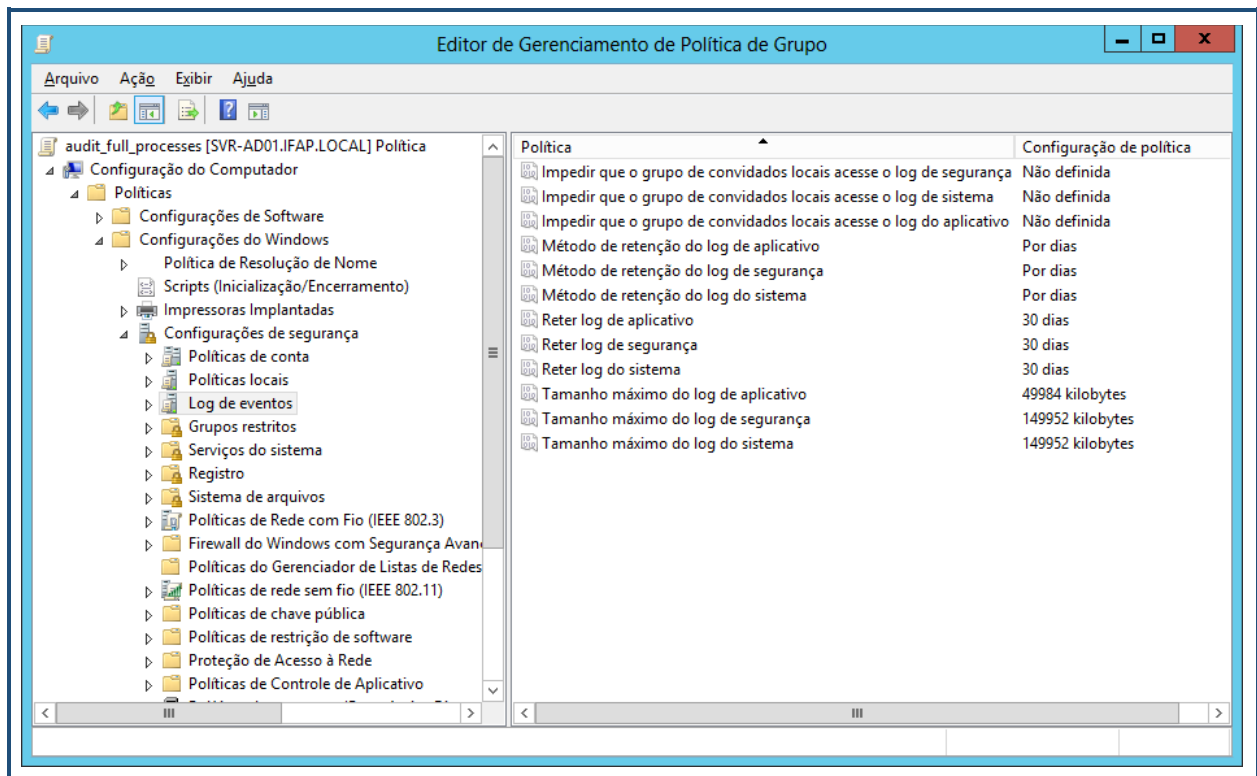


Fonte: Cenário real do IFAP – próprio autor

Há duas configurações nas diretivas de grupos que precisam ser definidas para que os eventos de segurança e de auditoria sejam registrados no servidor de origem dos *logs*. E a primeira delas é habilitar o método, o tempo e o tamanho de retenção dos *logs*.

- Na **ferramenta administrativa Gerenciamento de Diretiva de Grupo**, na **raiz da árvore de domínio**, **criar uma GPO** (*Group Policy Object* – objeto de política de grupo), para habilitar o registro dos *logs* de eventos;
- No módulo de **Configuração do Computador**, expandir a pasta **Configurações do Windows**, abrir as opções de **Configurações de Segurança** e, clicar no item **Log de eventos**;
- As diretivas que serão configuradas são referentes ao: Método de retenção do *log*; Tempo para Reter *log* e Tamanho máximo do *log*. Estas três diretivas são para os *logs* do tipo: Aplicativo; Segurança e Sistema;
- As configurações foram definidas como o demonstrado na Figura 9.

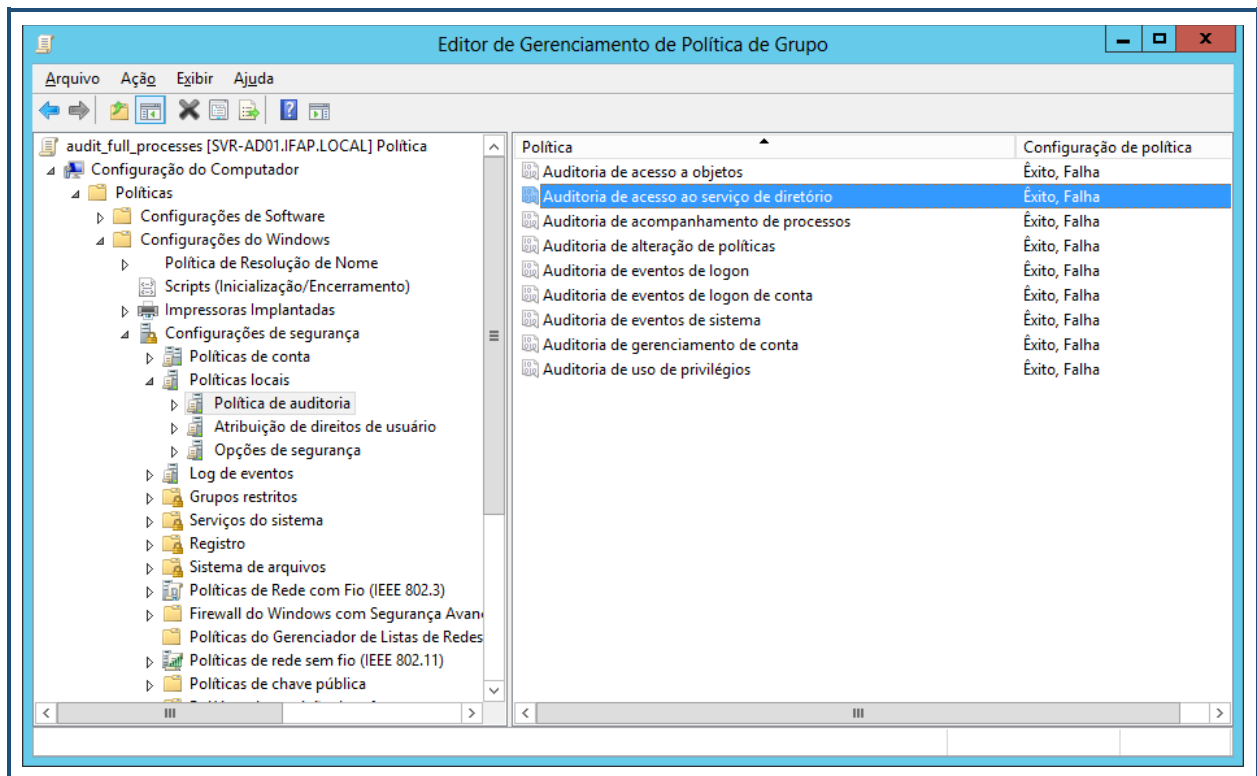
Figura 9: Configuração das diretivas de grupo – ativar *log* de eventos



Fonte: Cenário real do IFAP – próprio autor

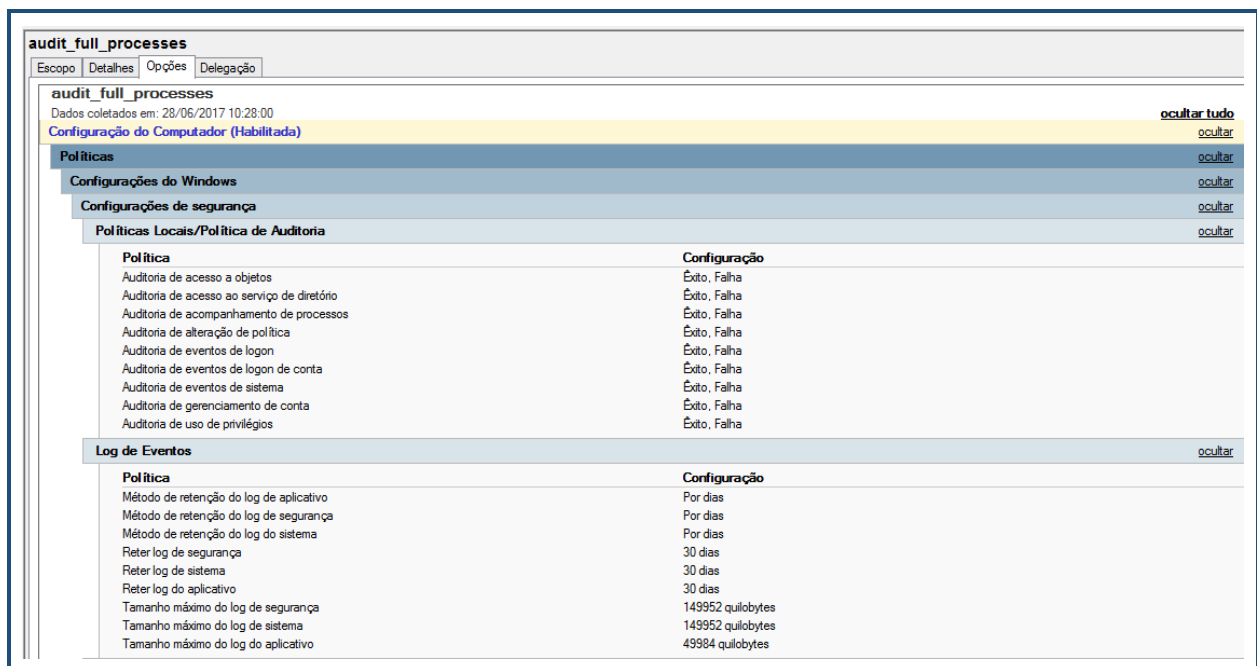
Há ainda mais uma configuração de diretivas de grupo a ser configura, é a diretiva que vai habilitar os registros de auditoria no *log* de eventos.

- Continuando na **ferramenta administrativa Gerenciamento de Diretiva de Grupo**, na **raiz da árvore de domínio**, editar a **mesma GPO** para habilitar as políticas de auditoria;
- No módulo de **Configuração do Computador**, expandir a pasta **Configurações do Windows**, abrir as opções de **Configurações de Segurança** e, expandir o item de **Políticas locais** e clicar sobre o item **Política de auditoria**;
- Habilitar todos os **eventos de auditoria**, para que registrem **ocorrências** quando ocorrer **êxito e falha**, conforme o apresentado na Figura 10:

Figura 10: Habilitar os registros de auditoria – êxito e falha

Fonte: Cenário real do IFAP – próprio autor

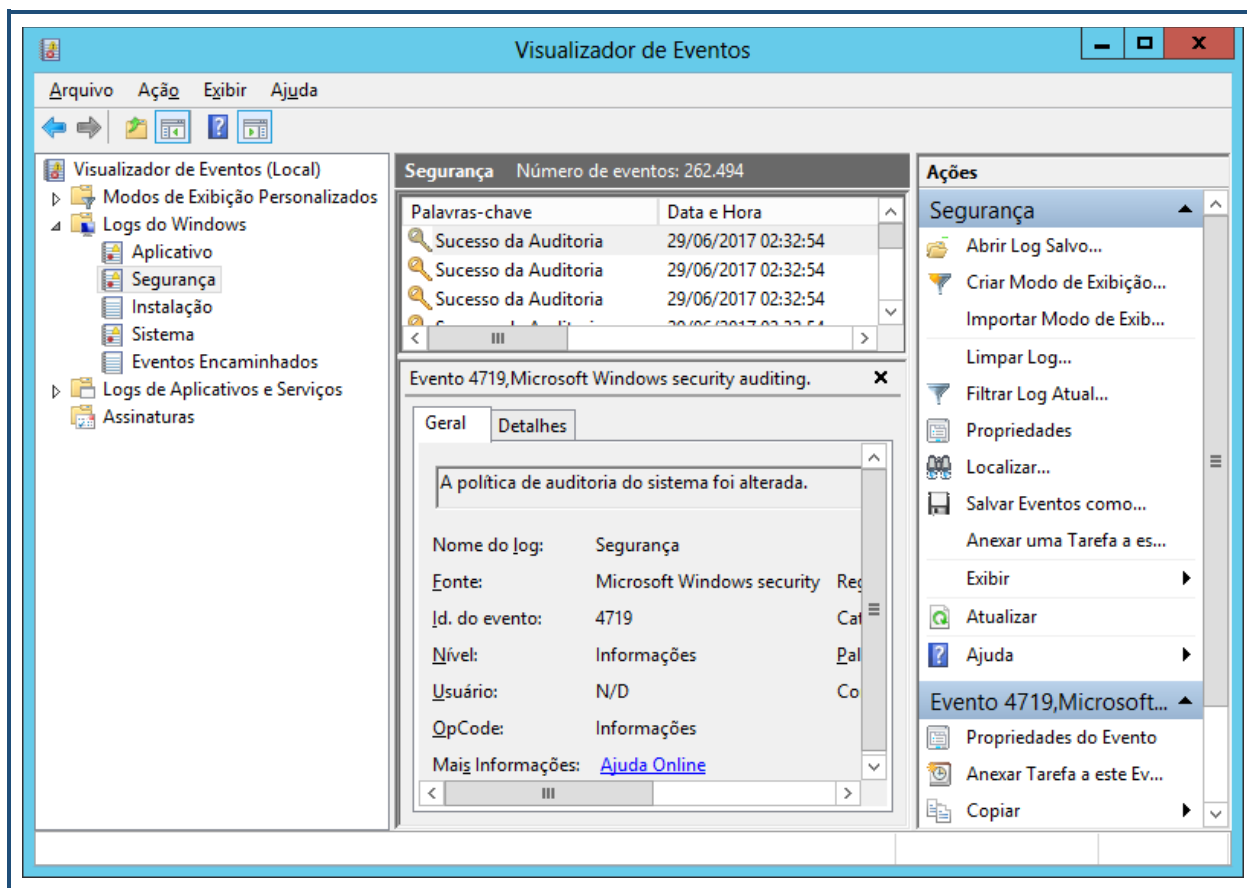
A série de diretivas de grupo (GPOs) que foram configuradas é o que possibilitará o registro dos Eventos de Auditoria e de Segurança, no arquivo de logs e, as configurações estão apresentadas na Figura 11.

Figura 11: Diretivas para o registro dos eventos de auditoria e de segurança

Fonte: Cenário real do IFAP – próprio autor

No painel central do Visualizador de eventos, no servidor de origem, pode-se visualizar os eventos sendo registrados, na Figura 12 estão apresentados os eventos de Segurança.

Figura 12: Visualizador de eventos no servidor origem – Eventos de Segurança



Fonte: Cenário real do IFAP – próprio autor

Pelo fato de estarmos encaminhando *logs* de eventos de segurança originados em um servidor controlador de domínio (AD), há a necessidade de se alterar as permissões para que a conta de Serviço de Rede tenha as permissões suficientes para o acesso de canal ao registro dos Eventos de Segurança do controlador do domínio.

Para alterar as permissões da conta do Serviço de Rede, deve-se executar, com credenciais de Administrador no servidor de domínio, no ambiente *Windows PowerShell*, o seguinte comando: `c:\wevtutil get-log security`. A informação que se busca, no resultado do comando, é o Identificador Seguro (SID – *Security Identifier*, pertencente ao registro dos Eventos de Segurança, para que a conta de Serviço de Rede acesse o canal seguro de comunicação, o Identificado está localizado no campo: `channelAccess`.

Quadro 5: Situação inicial do SID do registro dos Eventos de Segurança

```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. Todos os direitos reservados.

PS C:\Users\Administrador>wevtutil get-log security
name: security
enabled: true
type: Admin
owningPublisher:
isolation: Custom
channelAccess: O:BAG:SYD: (A;;0xf0005;;;SY) (A;;0x5;;;BA) (A;;0x1;;;S-1-5-32-573)
logging:
  logFileName: %SystemRoot%\System32\Winevt\Logs\security.evtx
  retention: false
  autoBackup: false
  maxSize: 153550848
publishing:
  fileMax: 1
```

Fonte: Cenário real do IFAP – próprio autor

Então o SID do canal de acesso do registro de Eventos de Segurança, necessita ser ajustado para que o Identificador Seguro (SID) da conta de Serviço de Rede do *Windows* seja adicionado ao SID do canal de acesso do registro de Eventos de Segurança. O Identificador de Segurança (SID) da conta de Serviço de Rede, por padrão no ambiente *Windows Server* é **S-1-5-20**, por isso é necessário adicioná-lo ao SDDL (*Security Descriptor Definition Language* – Linguagem de definição de descritor de segurança) como o demonstrado aqui usando o comando **wevtutil set-log** com o parâmetro **/ca** (channel access – acesso de canal) para dar a permissão de leitura para a conta de Serviço de Rede, no *Log* de Eventos de Segurança, a linha de comando e o resultado está descrita no Quadro 6.

Quadro 6: Inclusão do SID da conta de Serviço de Rede ao canal de acesso

```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. Todos os direitos reservados.

PS C:\Users\Administrador>wevtutil set-log security
/ca:'O:BAG:SYD: (A;;0xf0005;;;SY) (A;;0x5;;;BA) (A;;0x1;;;S-1-5-32-573) (A;;0x1;;;S-1-5-20)'
```

Fonte: Cenário real do IFAP – próprio autor

Após a inclusão do SID da conta de Serviço de Rede ao canal de acesso do registro de Eventos de Segurança, deve-se imediatamente verificar se a inclusão do SID foi bem sucedida, executando-se o comando do Quadro 7.

Quadro 7: Verificação do SID da conta de Serviço de Rede no canal de acesso

```

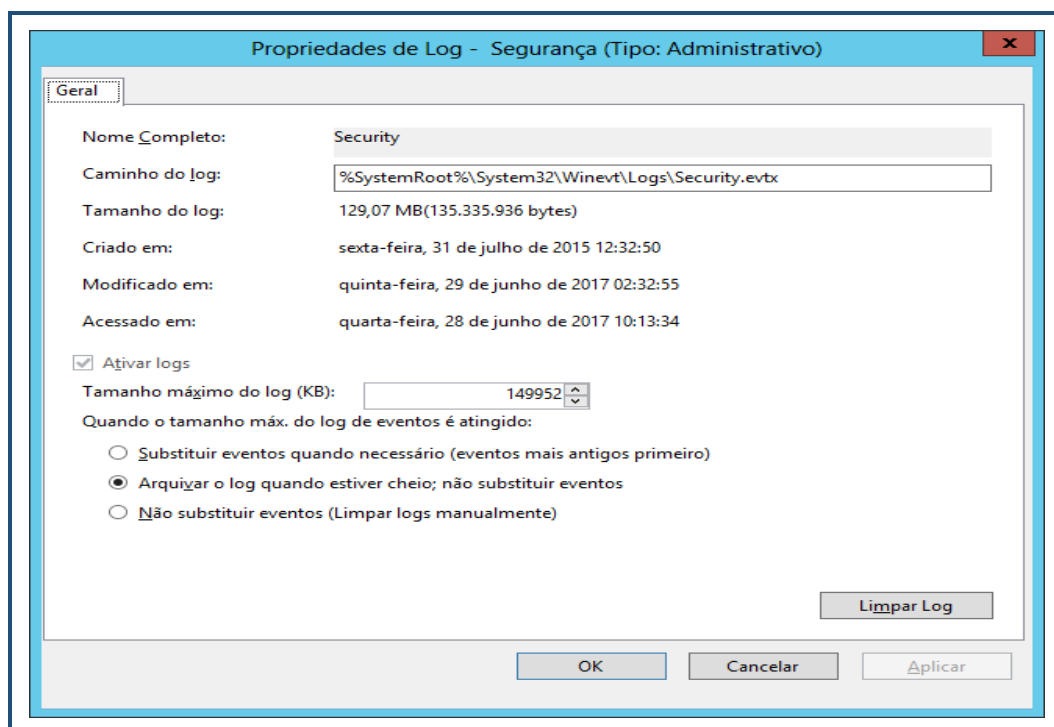
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. Todos os direitos reservados.

PS C:\Users\Administrador>wevtutil get-log security
name: security
enabled: true
type: Admin
owningPublisher:
isolation: Custom
channelAccess: O:BAG:SYD:(A;;0xf0005;;;SY)(A;;0x5;;;BA)(A;;0x1;;;S-1-5-32-
573)(A;;0x1;;;S-1-5-20)
logging:
  logFileName: %SystemRoot%\System32\Winevt\Logs\security.evtx
  retention: false
  autoBackup: true
  maxSize: 153550848
publishing:
  fileMax: 1

```

Fonte: Cenário real do IFAP – próprio autor

Para a confirmação das configurações da coleta dos Eventos de Segurança, pode-se verificar as configurações no Visualizador de Eventos do *Windows*, na máquina controlador de domínio. Onde serão mostradas as informações do caminho de armazenamento e arquivo de registro dos eventos de segurança, condição da ativação do *log*; tamanho máximo do arquivo de *log* e condição do rotacionamento do arquivo de *logs*.

Figura 13: Confirmação da ativação da coleta dos *logs* de Segurança

Fonte: Cenário real do IFAP – próprio autor

4.1.2 Configuração da Estação Coletora de Logs – *Windows 7 Professional* SP1

Com o servidor de origem de *log* de eventos, já configurado, parte-se para a configuração da estação coletora, neste processo duas etapas serão efetivadas: a) a primeira é configurar o modo

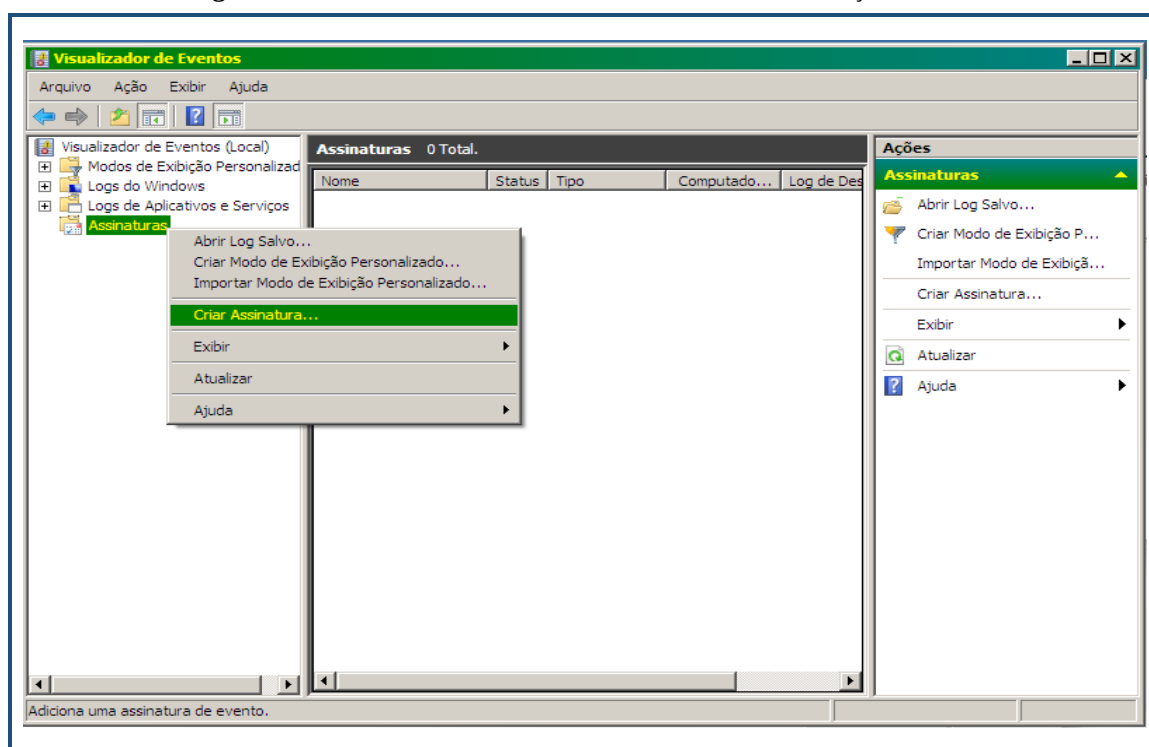
de inicialização do serviço Coletor de eventos do *Windows* e, b) a segunda é a criação de uma assinatura para o Servidor de origem (MICROSOFT, 2017)⁹.

a) Configurar uma assinatura no Visualizador de Eventos

A maneira mais fácil de se configurar uma assinatura do serviço do Coletor de Eventos, pela primeira vez, é utilizando a ferramenta **Visualizador de Eventos**:

- Abrir a opção **executar do menu Iniciar**, digitar **eventvwr.msc** e, **pressionar ENTER** para abrir o objeto do console de gerenciamento da *Microsoft*, **Visualizador de Eventos**;
- No console do **Visualizador de Eventos**, no painel da esquerda, clicar na Unidade Organizacional **Assinaturas**, Figura 14;
- Clicar no botão **Sim** na caixa de diálogo do **Visualizador de Eventos**, pois este procedimento inicializa o Serviço Coletor de Eventos do *Windows* e então o define para ser iniciado automaticamente no carregamento do sistema operacional, Figura 15.

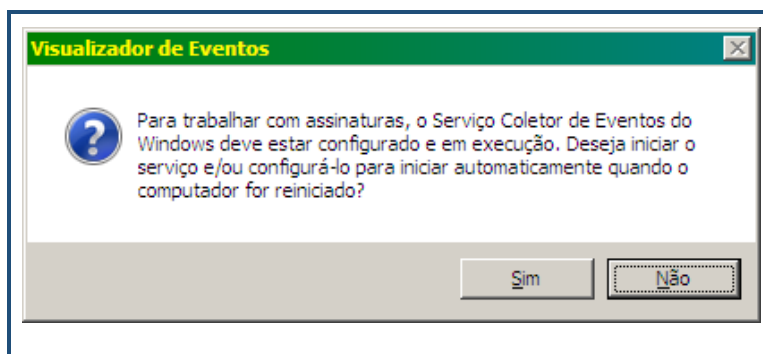
Figura 14: Criar assinatura no visualizador de eventos da estação coletora



Fonte: Cenário real do IFAP – próprio autor

⁹ Criar assinatura: [https://technet.microsoft.com/pt-br/library/cc722010\(v=ws.11\).aspx](https://technet.microsoft.com/pt-br/library/cc722010(v=ws.11).aspx)

Figura 15: Início do Serviço Coletor de Eventos, com inicialização automática



Fonte: Cenário real do IFAP – próprio autor

Este procedimento pode ser realizado, também, a partir da linha de comando do *Windows*, com credenciais de Administrador, digitando o comando: `c:\wecutil qc`

Quadro 8: Inicializar o serviço coletor de eventos do *Windows*, na estação coletora

```
Microsoft Windows [versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\luiz-hamilton.silva>wecutil qc
O modo de inicialização do serviço será alterado para Delay-Start (Atrasar Início).
Deseja continuar (S- sim ou N- não)? S

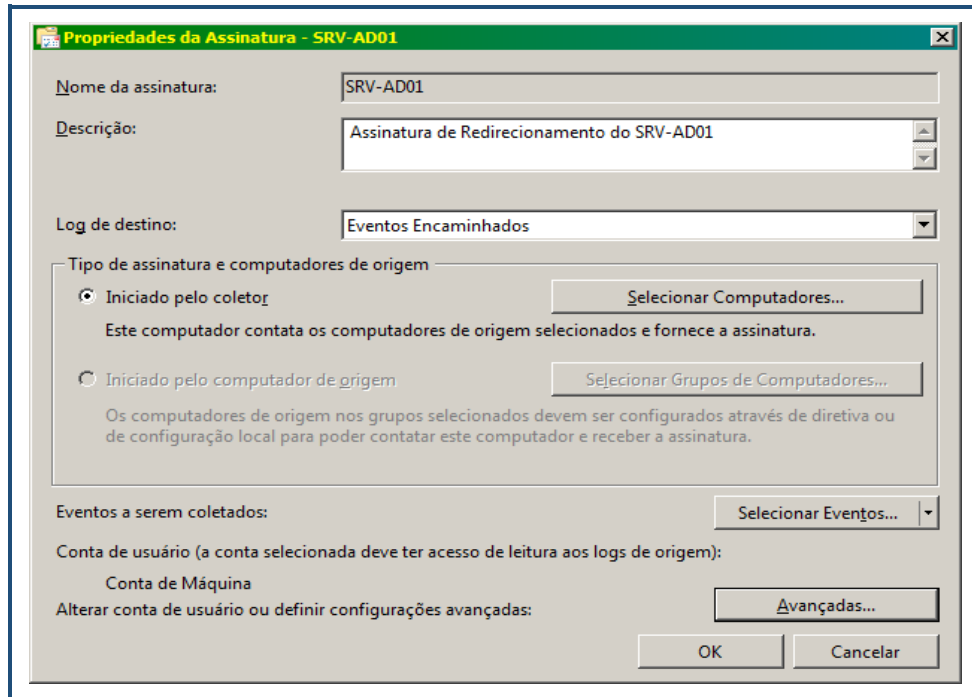
O serviço Coletor de Eventos do Windows foi configurado com êxito.
```

Fonte: Cenário real do IFAP – próprio autor

Após a ativar e confirmar a inicialização automática do serviço coletor de eventos do *Windows*, o procedimento seguinte é criar uma assinatura de coleta, a qual irá consultar os registros de *logs* de eventos no Servidor de origem, que foi previamente configurado na Seção 4.1.1.

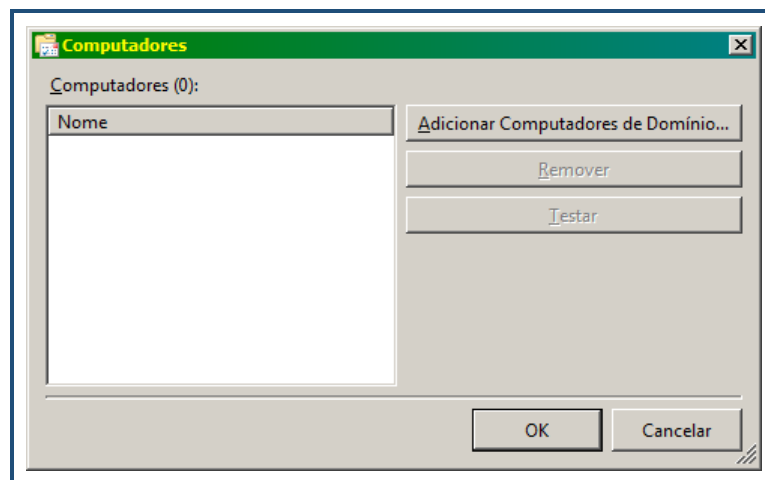
- Ainda na caixa de diálogo do **Visualizador de Eventos**, no painel **Ações** à direita, clicar em **Criar assinatura**;

Nas Figuras 16 e 17, são apresentadas as telas de configuração para se criar a assinatura para a estação que atuará no papel de coletor de eventos. Então a sequência de procedimentos é a seguinte:

Figura 16: Configurar uma assinatura no Visualizador de Eventos

Fonte: Cenário real do IFAP – próprio autor

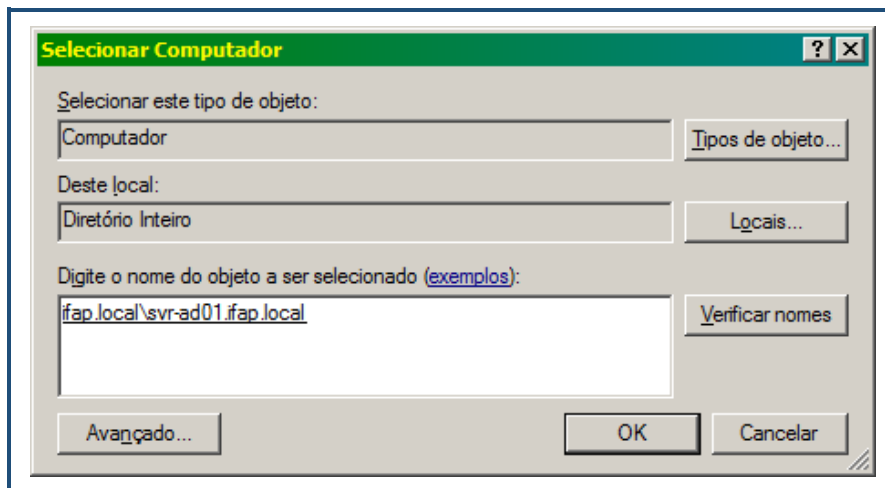
- Na caixa de diálogo **Propriedades de Assinatura**, campo **Nome da assinatura** dar um nome personalizado à assinatura de coleta;
- Ainda na caixa de diálogo **Propriedades de Assinatura**, no campo **Descrição** dar uma descrição à assinatura de coleta;
- No menu suspenso **Log de destino**, manter a opção **Eventos Encaminhados**, que é a Unidade Organizacional onde serão salvos os eventos a serem coletados;
- Na seção **Tipo de assinatura e computadores de origem**, certificar-se que a opção **Iniciado pelo coletor**, esteja ativa e clique no botão **Selecionar Computadores**;

Figura 17: Adicionar computadores do Domínio

Fonte: Cenário real do IFAP – próprio autor

- Na caixa de diálogo **Computadores**, clicar em **Adicionar computadores de domínio**;
- Na caixa de diálogo **Selecionar Computador**, digite o nome do computador de origem dos eventos e, na seção **Digite o nome do objeto a ser selecionado**, então clique no botão **Verificar nomes** e, então clicar em **OK**, para confirmar, Figura 18;

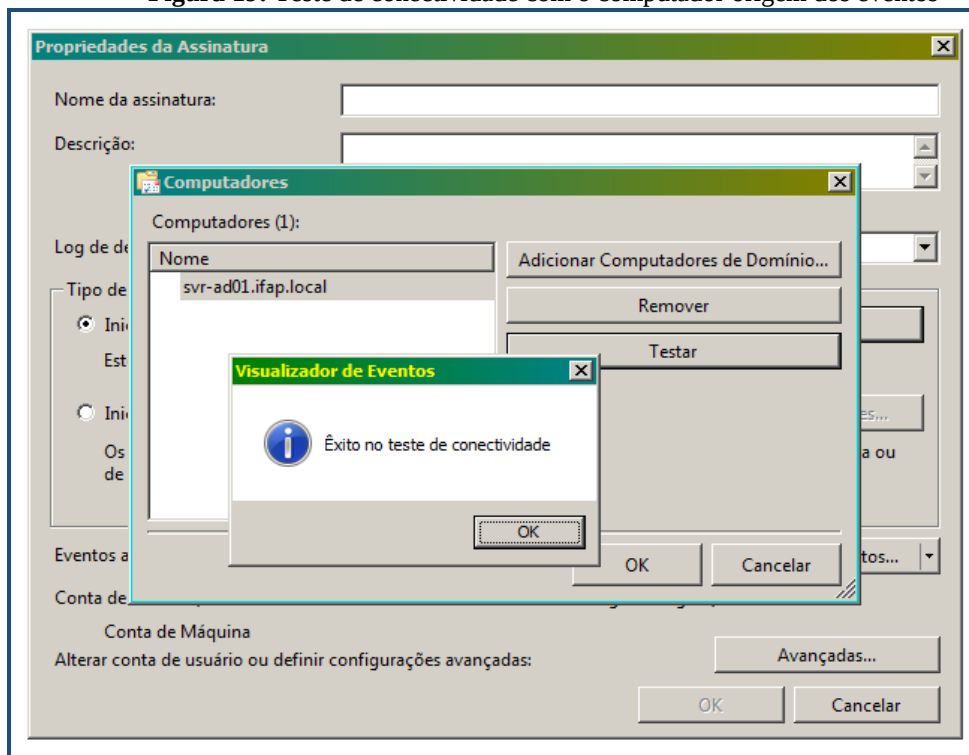
Figura 18: Selecionar computador origem dos eventos – assinatura de coleta



Fonte: Cenário real do IFAP – próprio autor

- E finalizando a seleção, o nome do computador de origem aparecerá na caixa de diálogo **Computadores**. Então, verificar se ele está selecionado, no painel à esquerda e clicar no botão **Testar**.

Figura 19: Teste de conectividade com o computador origem dos eventos



Fonte: Cenário real do IFAP – próprio autor

Se tudo foi corretamente configurado, até este momento, o sistema emitirá a mensagem de que o teste de conectividade foi bem-sucedido. Isso significa que o computador coletor será

capaz de se conectar ao computador origem dos eventos, utilizando o serviço WinRM;

- Clicar em OK na caixa de diálogo **computadores**, Figura 17.
- Ainda na caixa de diálogo **Propriedades da Assinatura**, na seção **Eventos a serem coletados**, clicar no botão **Selecionar Eventos**.

O procedimento seguinte é criar um filtro de pesquisa para a Assinatura, indicando o **Nível do evento** e a **Fonte de evento**, Figura 20.

Figura 20: Criar um filtro dos logs de Eventos no coletor

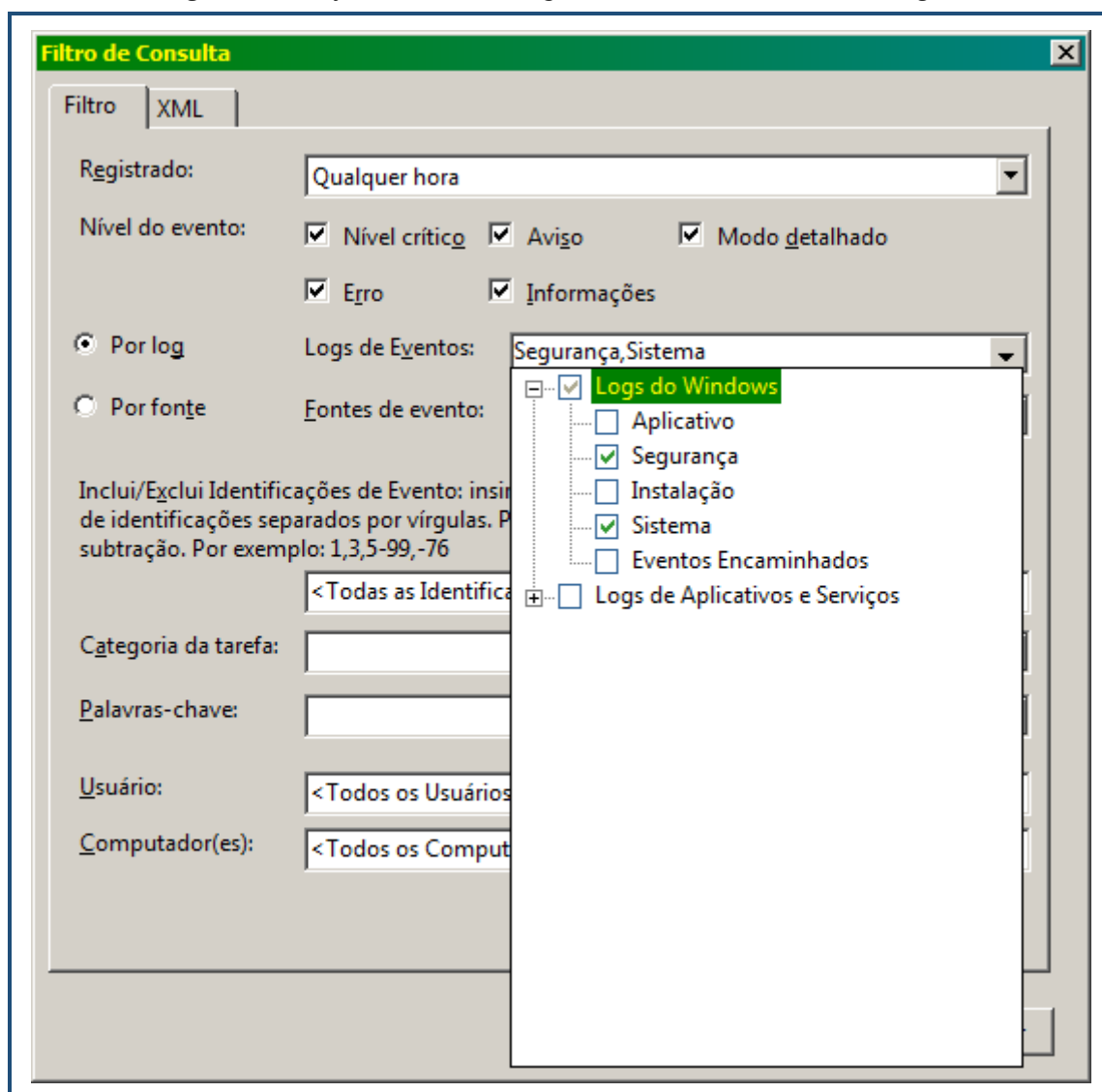
Fonte: Cenário real do IFAP – próprio autor

- Na caixa de diálogo **Filtro de consulta** na guia **Filtro**, marcar as caixas de verificação do **Nível de evento**, Nível crítico; Aviso; Modo detalhado; Erro; Informações;
- Marcar **Por log** e, em seguida à direita, clicar no menu suspenso **Logs de Evento**;
- No menu suspenso que se desdobra, marcar os *Logs* de interesse da coleta, aqui foi selecionado: **Segurança**;

Na barra de rolagem se tem acesso à lista de todos os *logs* possíveis de serem coletados a partir do servidor origem. Nas Figura 21, são apresentadas as seleções possíveis dos *logs* de

interesse que podem ser coletados a partir servidor origem dos eventos:

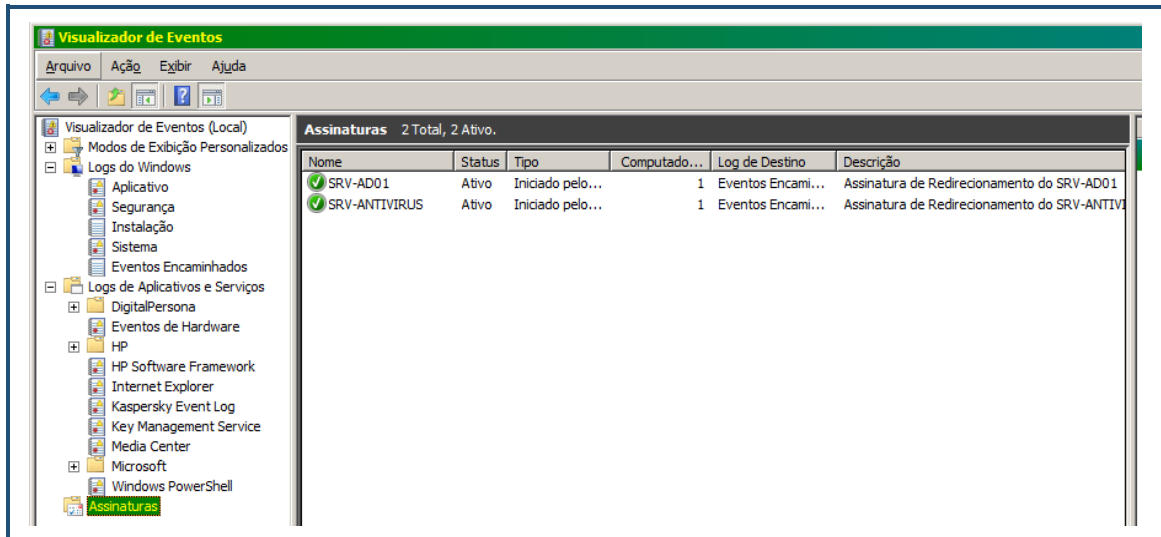
Figura 21: Seleção dos filtros dos logs a serem coletados do servidor origem



Fonte: Cenário real do IFAP – próprio autor

- Clicar em uma **área neutra** da janela **Filtro de consulta**. Então clicar em **OK** para fechar a caixa de diálogo Filtro de consulta;
- E finalmente, clicar em **OK** na caixa de diálogo **Propriedades de assinatura** para concluir o processo de seleção dos tipos de eventos a serem coletado da origem.

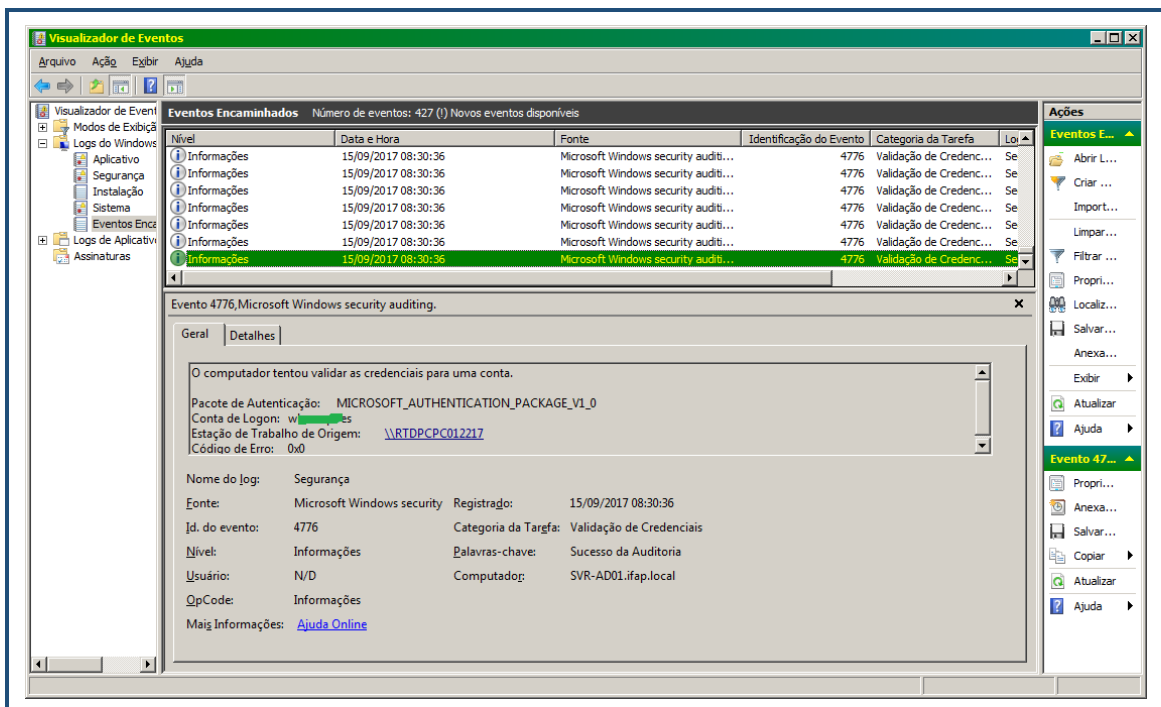
Figura 22: Lista das assinaturas criadas para a coleta de eventos



Fonte: Cenário real do IFAP – próprio autor

Aguarda-se a replicação dos serviços WinRM entre a estação coletora e o servidor originador dos *logs*, o que neste cenário levou uns 30 segundos para o início da coleta dos *logs* de Segurança provenientes do Servidor SVR-AD01. Então os *logs* começam a ser coletados e, visualizados na seção Eventos Encaminhados, conforme o demonstrado na Figura 23.

Figura 23: Visualizador de Eventos na estação coletora – *logs* de Segurança

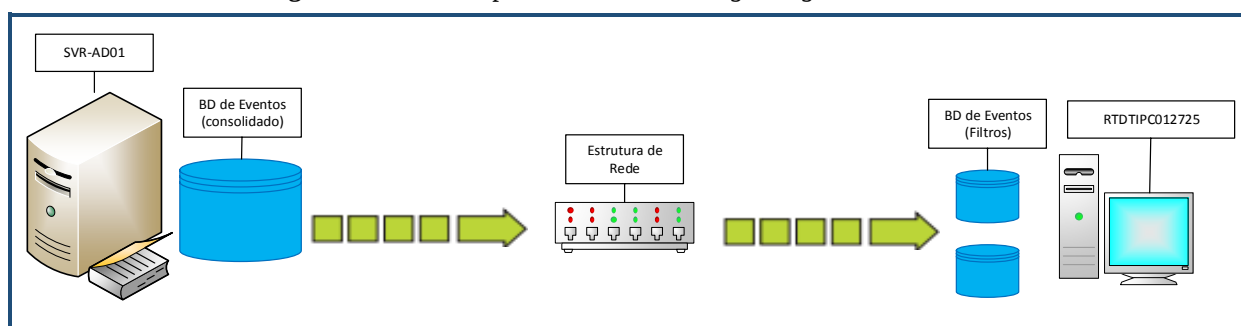


Fonte: Cenário real do IFAP – próprio autor

4.2 Método Proposto para a Análise e Auditoria dos Eventos de Segurança

O objetivo maior desta pesquisa é realizar a análise de *logs* de eventos com o propósito de se identificar os eventos relevantes para os procedimentos de auditoria em ocorrências conexas às contas de usuários. A abordagem da coleta e filtragem dos arquivos de eventos registrados no servidor originador dos *logs* está baseada no modelo *loghost* centralizado e, para compreender-se a dinâmica do o fluxo de captura e armazenamento dos *logs* dos Eventos de Segurança, a abstração deste fluxo é apresentada na Figura 24.

Figura 24: Modelo aplicado à coleta dos *logs* – *loghost* centralizado



Fonte: Elaborado pelo autor

A técnica adotada é gerar *Scripts* .PS1 para a execução no ambiente *PowerShell*, compatíveis com todas as versões dos sistemas operacionais *Windows* que disponham deste ambiente de linha de comando. Para a tomada de decisão de quais eventos serão apresentados como modelo específico nesta pesquisa, foi consultada a recomendação da (MICROSOFT, 2017)¹⁰ de quais eventos devam ser monitorados no serviço de autenticação (*Active Directory*), para se garantir o não comprometimento deste serviço. E, também, se consultou outras recomendações para o processo de auditoria dos registros de eventos, em ambiente *Windows* (MICROSOFT, 2017)¹¹.

Então definiu-se 05 identificadores relacionados aos Eventos de Segurança, os quais ocorrem em qualquer ambiente do serviço de *Active Directory*, que servirão de modelo para a análise e auditoria nesta pesquisa e, estes eventos escolhidos são de grande interesse nas atividades diárias do Gerenciamento de Segurança em um ambiente de Rede, os quais estão elencados na Tabela 10.

¹⁰ Monitorar Eventos do AD: [https://technet.microsoft.com/pt-br/library/dn535498\(v=ws.11\).aspx](https://technet.microsoft.com/pt-br/library/dn535498(v=ws.11).aspx)

¹¹ Auditoria: <https://blogs.technet.microsoft.com/sooraj-sec/2016/08/20/logparser-play-of-a-forensicator/>

Tabela 10: Eventos em análise nos registros dos *Logs* de Segurança

ID do evento	finalidade
4624	Localizar e identificar as contas de usuários que efetuaram acesso remoto (RDP – <i>Remote Desktop Connection</i>).
4625	Localizar e identificar usuários que não obtiveram sucesso ao tentar autenticar-se, ou ainda, uma possível tentativa de uso de credenciais alheias.
4648	Localizar e identificar as contas de usuários ou serviços que estão realizando acesso com credenciais explícitas, quando o usuário ou serviço utiliza outra conta para acessar recursos específicos.
4660 e 4663	Localizar e identificar objetos excluídos por usuários (monitorando o compartilhamento de arquivos).
4771	Localizar e identificar quais usuários, naquele momento, estão com a conta bloqueada após falhas de <i>logon</i> , devido ao fornecimento de credenciais incorretas.

Fonte: Eventos de Segurança (MICROSOFT, 2017)¹²

Para a construção das *queries* SQL, há a necessidade de se conhecer a estrutura dos campos de um arquivo de *logs* de eventos nos sistemas *Windows* e em seu formato original de arquivo de eventos, formato EVTX, essa estrutura de campos é apresentada na Tabela 11.

Tabela 11: Estrutura de arquivo de *log* de eventos nos sistemas *Windows* (EVTX)

campo	descrição
<i>EventLog</i>	Nome do arquivo do registro de Eventos.
<i>RecordNumber</i>	O número do registro no banco de dados de Eventos.
<i>TimeGenerated</i>	O momento em que a entrada foi enviada. Este tempo é medido em número de segundos decorridos desde 00:00:00 de 1º de janeiro de 1970, no formato de horário UTC.
<i>TimeWritten</i>	O momento em que a entrada foi recebida pelo serviço para ser escrita no registro. Este tempo é medido em número de segundos decorridos desde 00:00:00 de 1º de janeiro de 1970, no formato de horário UTC.
<i>EventID</i>	O valor do ID é específico do originador do evento para o registro do evento e é usado com o nome de origem para localizar uma <i>string</i> de descrição no arquivo de mensagem para o originador do evento ($priority = facility * 8 + severity$)
<i>EventType</i>	O tipo de Evento, que pode assumir uma série de Tipos: 0x0001 = <i>Error event</i> 0x0010 = <i>Failure Audit event</i> 0x0008 = <i>Success Audit event</i> 0x0004 = <i>Information event</i> 0x0002 = <i>Warning event</i>

¹² Lista de Eventos de Segurança: <https://support.microsoft.com/en-us/help/977519/description-of-security-events-in-windows-7-and-in-windows-server-2008>

<i>EventTypeName</i>	Identifica os ID de Eventos por nomes: EVENTLOG_ERROR_TYPE = 0x0001 EVENTLOG_AUDIT_FAILURE = 0x0010 EVENTLOG_AUDIT_SUCCESS = 0x0008 EVENTLOG_INFORMATION_TYPE = 0x0004 EVENTLOG_WARNING_TYPE = 0x0002
<i>EventCategory</i>	A categoria para o Evento. O significado desse valor depende da fonte do evento.
<i>EventCategoryName</i>	Nome da categoria gerado para o evento. Depende da fonte do evento.
<i>SourceName</i>	Nome da fonte geradora do Evento.
<i>Strings</i>	Uma sequência de caracteres de descrição dentro deste registro específico, em referência ao de registro de eventos. Composto por informações do módulo gerador do evento e, acompanhado da identificação do originador do evento de ID do evento.
<i>ComputerName</i>	Nome da estação de onde se gerou o Evento.
<i>SID</i>	SID (Identificador de Segurança) da conta de usuário ou de serviço que provocou o Evento.
<i>Message</i>	Gerado a partir dos dados da seção <i>Strings</i> e, de informações contidas dentro de DLLs do sistema de origem.
<i>Data</i>	Dados específicos do Evento. Esta informação pode ser algo específico (um driver de disco pode registrar o número de tentativas, por exemplo), seguido de informações binárias específicas para o evento que está sendo registrado e para a origem que gerou a entrada.

Fonte: Adaptado de (MICROSOFT, 2017)¹³

A análise do arquivo de *logs* com a ferramenta *Microsoft Log Parser 2.2*, no ambiente de linha de comando (CLI), utiliza expressões de consulta SQL (*Structured Query Language* – Linguagem de Consulta Estruturada) e, para se extrair informações do arquivo de *logs* necessita-se lançar mão de sintaxes específicas, incluindo além da informação do arquivo de *logs*, com seu caminho completo, dados específicos dos campos de consulta e tipos de dados que quer extrair e, um exemplo de sintaxe SQL dentro do ambiente CLI da ferramenta *Microsoft Log Parser 2.2* é apresentado na Figura 25.

¹³ Estrutura de Eventos: [https://msdn.microsoft.com/pt-br/library/windows/desktop/aa363646\(v=vs.85\).aspx](https://msdn.microsoft.com/pt-br/library/windows/desktop/aa363646(v=vs.85).aspx)

Figura 25: Consulta SQL ao arquivo de logs, na ferramenta de linha de comandos

```

C:\Program Files (x86)\Log Parser 2.2>logparser -i:EVT "SELECT TimeGenerated, SourceName,
Message INTO resultados_bloqueio.txt FROM 'C:\disserta\LPSV2.D1-LH\logs\Security\Security_
ORIGINAL_EN.evtx' WHERE STRINGS LIKE '%06A%'
WARNING: Output format not specified - using NAT output format.

Statistics:
-----
Elements processed: 278036
Elements output:    1241
Execution time:     8.40 seconds

C:\Program Files (x86)\Log Parser 2.2>

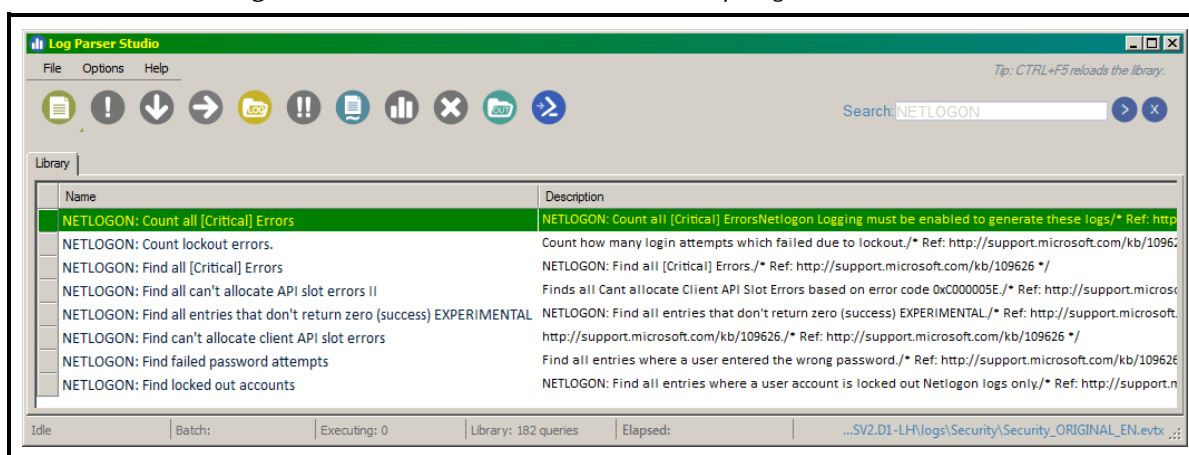
```

Fonte: Cenário real do IFAP – próprio autor

A ferramenta *Log Parser 2.2* que é um analisador de logs, apresentando-se como uma ferramenta poderosa e versátil que fornece acesso de consulta universal a dados baseados em texto, como arquivos de log, arquivos XML e arquivos CSV, bem como fontes de dados importantes no sistema operacional *Windows*, como o Registro de Eventos, as chaves de Registro, o sistema de arquivos e a estrutura de diretório do *Active Directory* (MICROSOFT, 2017)¹⁴.

A outra ferramenta em comparação nesta pesquisa é a *Microsoft Log Parser Studio 2.0* e, as características fortes desta ferramenta são: seu ambiente gráfico (GUI), apresentando recursos de interação com diversos formatos de arquivos de dados, além da funcionalidade de se exportar as consultas desenvolvidas em consultas SQL, para o formato dos scripts PS1 (*PowerShell*). Esta ferramenta apresenta a aba *Library*, onde estão contidas algumas queries SQL prontas para algumas ações relativas às consultas de arquivos de registros, como os provenientes de sistemas *MS-Exchange*; *MS-ActiveSync*; *MS-Internet Information Service*, dentre outras fontes de dados e, é importante registrar que na biblioteca disponibilizada pela ferramenta LPS 2.0, não há disponíveis queries prontas para os eventos propostos para análise nesta pesquisa, tela inicial na Figura 26.

Figura 26: Tela inicial da ferramenta *Microsoft Log Parser Studio 2.0*

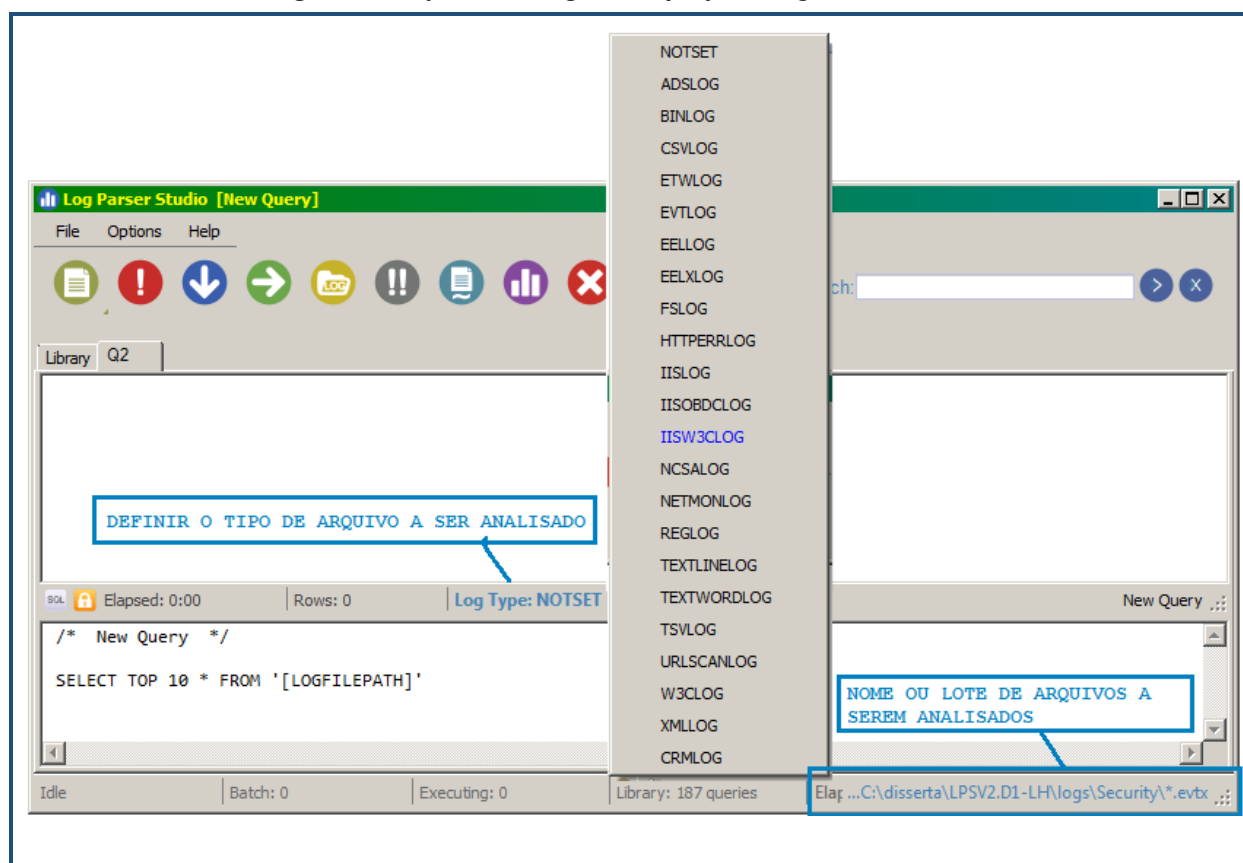


Fonte: Cenário real do IFAP – próprio autor

¹⁴ *Log Parser 2.2*: <https://technet.microsoft.com/en-us/scriptcenter/dd919274.aspx>

Na janela de criação de *queries* personalizadas a ferramenta LPS 2.0 apresenta duas seções a parte de cima da janela apresentará os resultados consultados, enquanto a parte de baixo é utilizada para se escrever as consultas em linguagem SQL, com o detalhe que a ferramenta exige a especificação do formato do arquivo que está sob análise e a indicação do caminho e nome do arquivo que será alvo das consultas, como o mostrado na Figura 27.

Figura 27: Seção de montagem das *queries* no *Log Parser Studio 2.0*



Fonte: Cenário real do IFAP – próprio autor

Na apresentação da ferramenta *Log Parser Studio 2.0*, no sítio da *Microsoft* esta ferramenta é apresentada como sendo criada para atender à lacuna da sua ferramenta anterior – *Log Parser 2.2*, oferecendo interface gráfica, o que permite aos usuários da ferramenta uma forma mais rápida e eficiente de se obter dados sem “peripécias”, a partir do uso de *scripts* de consulta (MICROSOFT, 2017)¹⁵.

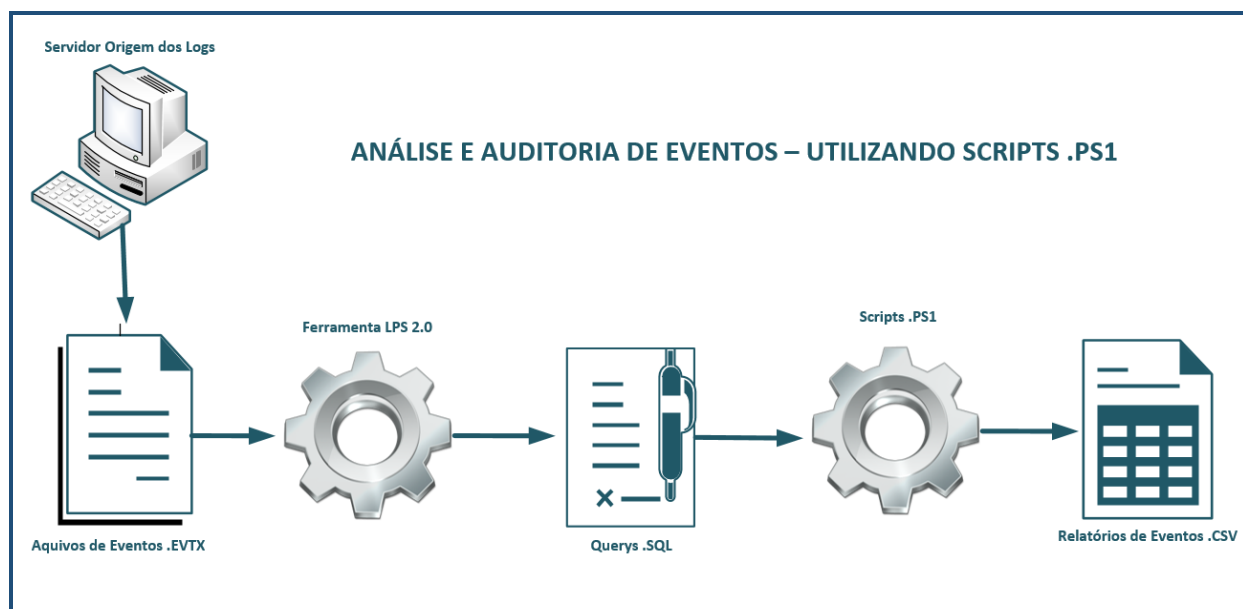
Essas duas ferramentas que estão em comparação, separam-se pela forma do ambiente da análise, uma usa o ambiente de linha de comando (CLI) e, a outra utiliza ambiente gráfico (GUI) e, justamente pelo fato da ferramenta *Log Parser Studio 2.0*, possuir a função de conversão das sintaxes SQL para o formato de *scripts PowerShell*, foi a ferramenta escolhida para a construção das consultas SQL e para a produção do ferramental de *scripts PS1*.

¹⁵ *Log Parser Studio*: <https://blogs.technet.microsoft.com/exchange/2012/03/07/introducing-log-parser-studio/>

Para o processamento das consultas às informações do arquivo de registros de eventos capturado pela estação coletora de *logs*, o volume de dados capturados foi exportado para arquivo *off-line*, o que facilita o processamento da ferramenta analisadora de *logs*, pelo fato de não trabalhar com arquivo em produção. E, embora os sistemas *Windows* disponibilizem nativamente uma ferramenta de visualização de eventos, os dados constantes nos registros de eventos têm enorme riqueza de informações, as quais não são de fácil leitura ao se utilizar somente a ferramenta Visualizador de Eventos do *Windows* e, se torna uma tarefa não tão clara e, às vezes inexistente consultas próprias ou personalizáveis para se encontrar informações específicas dentro do arquivo que se está analisando e, por isso é altamente recomendável recorrer às ferramentas analisadoras de *logs*, algumas são de uso e de código livres e outras proprietários.

O método proposto nesta pesquisa centra-se em oferecer ao Administrador de Redes nos sistemas operacionais *Windows*, um ferramental de interação com os arquivos de registro de eventos e, em particular o cenário de testes desta pesquisa foi executado sobre o arquivo de registro de Eventos de Segurança (*Security.evtx*) de um sistema *Windows Server 2012* que exerce o papel de servidor de autenticação (AD) e, o método proposto para a análise e auditoria dos registros coletados, visa a identificação de eventos relativos ao Serviço de Autenticação (*Active Directory*). A sequência de ações do método está descrita na Figura 28.

Figura 28: Modelo proposto para a análise e auditoria dos registros de eventos



Fonte: Elaborado pelo autor

Neste modelo proposto, os eventos que se determinou para a análise, os quais estão explicitados na Tabela 10 serão, primeiramente, testados via *sintaxe* SQL utilizando a ferramenta LPS 2.0, onde se pode depurar a construção das *querys* SQL e, analisar se os resultados são os esperados, registre-se que se está utilizando um arquivo *stand-alone*, uma réplica do arquivo

original *Security.evtx*. Então, ocorrendo o sucesso na construção da sintaxe SQL, utiliza-se o recurso da própria ferramenta LPS, para se exportar as *querys* construídas para o formato de *scripts* PS1 (*PowerShell*) e, com estes *scripts* resultantes consolida-se no ferramental que o Administrador de Rede poderá utilizar em suas atividades diárias, como um recurso de análise e auditoria dos registros de eventos que, aqui especificamente são os Eventos de Segurança. Ressaltando-se que as análises propostas na Tabela 10 não existem, nativamente, dentro de nenhum Sistema Operacional Windows (MICROSOFT, 2017)¹⁶.

¹⁶ Interação *LogParser* e *PowerShell*: <https://blogs.technet.microsoft.com/sooraj-sec/2017/02/20/logparser-and-powershell-logpower/>

5

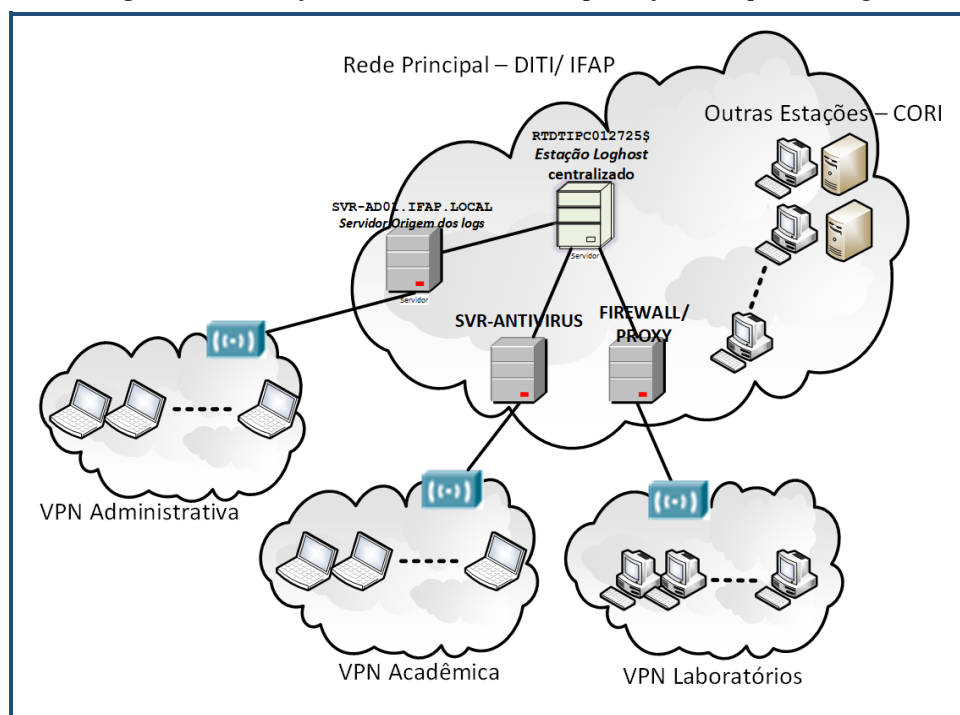
ANÁLISE DOS RESULTADOS

Neste capítulo são apresentados os resultados alcançados, dentro dos limites do ambiente de aplicação do modelo de análise de *logs* para auditoria de registro de Eventos de Segurança, utilizando o método proposto, aplicando o ferramental desenvolvido, que são os *scripts* PS1 (*PowerShell*), pré-configurados para as funções específicas das análises propostas e, alcançando a efetivação de ações que demonstrem os eventos conexos ao Serviço de Autenticação (*Active Directory*), que são a base para o procedimento diário de Gerenciamento da Segurança em um ambiente de Rede.

5.1 Ambiente de Testes

Em relação ao ambiente lógico dos computadores na rede, a abstração do ambiente é o mostrado na Figura 29, ressaltando que o ambiente analisado por (TSUNODA; *et al.*, 2009) guarda enorme similaridade com o ambiente real do IFAP, que é o ambiente em análise nesta pesquisa.

Figura 29: Abstração do ambiente real de implantação da captura de *logs*

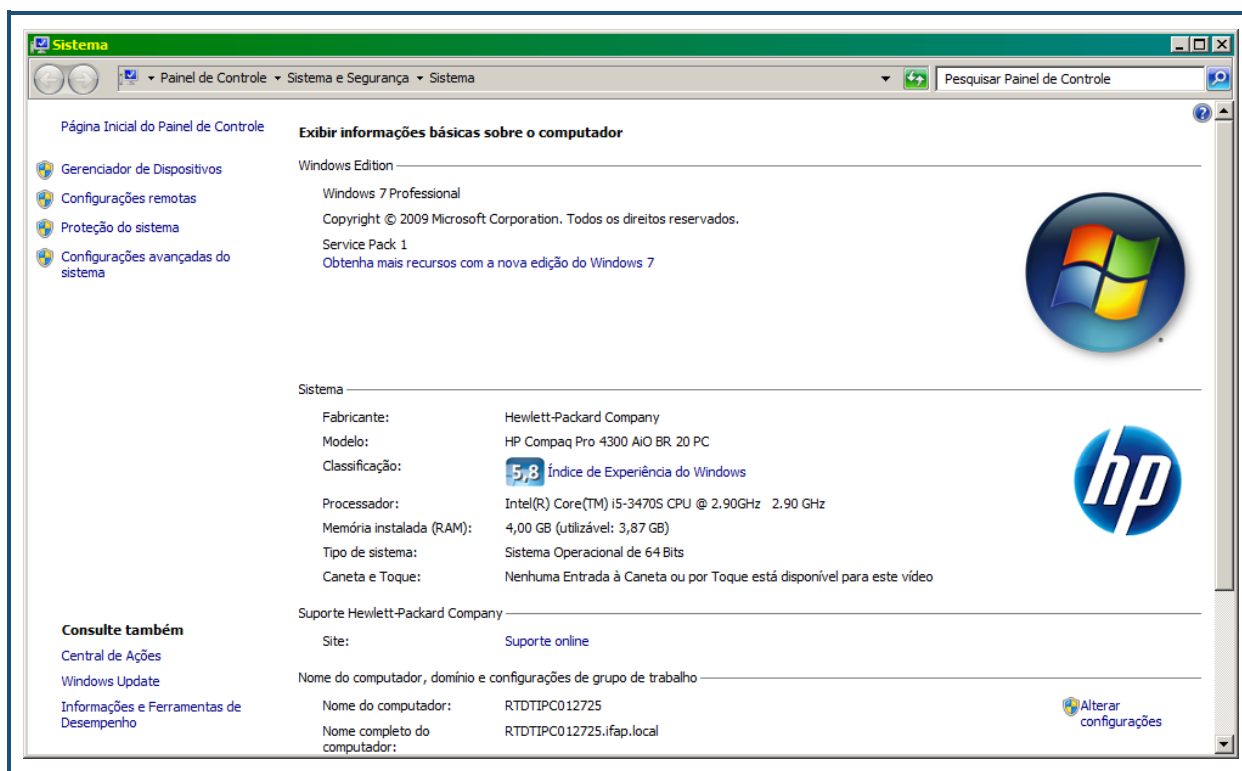


Fonte: Adaptado de (TSUNODA; *et al.*, 2009)

5.1.1 Detalhes das Configurações

Na estação coletora: RTDTIPC012725 os *logs* provenientes do servidor encaminhador de *logs*: SVR-AD01, foram armazenados e rotacionados, tendo como limite o tamanho máximo definido para o arquivo de *logs* dos Eventos de Segurança (*Security.evtx*) que no ambiente desta pesquisa foi estabelecido em 150 *Megabytes*, justificando-se essa escolha de tamanho dos arquivos de *logs*, para que os sistemas geradores de *logs* não necessitem despendar grande parte de seus recursos ao gerenciarem enormes arquivos de *logs* e, para que o tempo de análise a ser dedicado pelas ferramentas analisadoras de *logs* em relação ao volume do registro de eventos seja tolerável e não oneroso ao processo de Gerência de Segurança e, conseqüentemente, o tamanho dos arquivos gerados impactará em um tempo menor de processamento a ser utilizado pelos *scripts* PS1 (*PowerShell*). No tocante à análise de *logs*, optou-se pelo uso da ferramenta *Microsoft Log Parser Studio 2.0*, pelo fato de possuir interface gráfica e o recurso de gerar os *scripts* PS1 a partir das sintaxes das *queries* SQL

Figura 30: Recursos de processamento da estação coletora



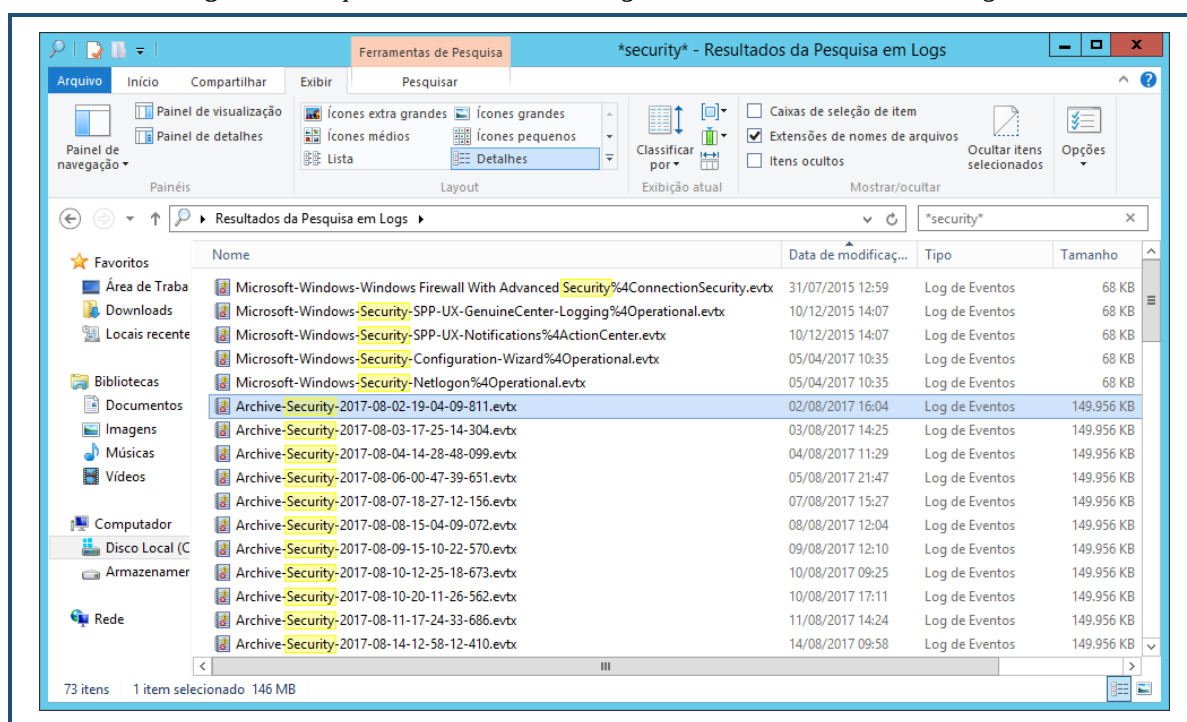
Fonte: Cenário real do IFAP – próprio autor

Os registros dos eventos de segurança remanescentes no servidor SVR-AD01, subsistem desde o dia 31 do mês de julho do ano de 2015, até o dia 29 do mês de junho de 2017, totalizando mais 750 dias de coleta, mas a totalidade destes eventos coletados não reúnem todas as informações importantes para a análise de auditoria, uma vez que as configurações do que os eventos retinham de informações não estavam definidas, pois o registro de eventos por padrão no

Windows não guardam informações relevantes para os procedimentos de auditoria, o que só ocorreu com a intervenção desta pesquisa. Portanto a análise dos eventos de segurança centra-se nos registros realizados a partir das configurações propostas nesta pesquisa – detalhe das configurações na Figura 11 – o que só ocorreu a partir de 28 de junho de 2017.

No ambiente específico desta pesquisa estão sendo filtrados os *logs* dos Eventos de Segurança, que no servidor origem: SVR-AD01, são armazenados no caminho: %SystemRoot%\System32\Winevt\Logs\, que é o arquivo em produção antes de ocorrer o arquivamento de *logs*, com base no tamanho de arquivos definido para que ocorra o rotacionamento. E o conjunto de registro dos Eventos de Segurança estão divididos em vários arquivos, nos tamanhos definidos para o rotacionamento, que foi de 150 *Megabytes* e, parte da lista de arquivos dos *logs* dos eventos de segurança, armazenados no servidor origem: SVR-AD01, é mostrada na Figura 31.

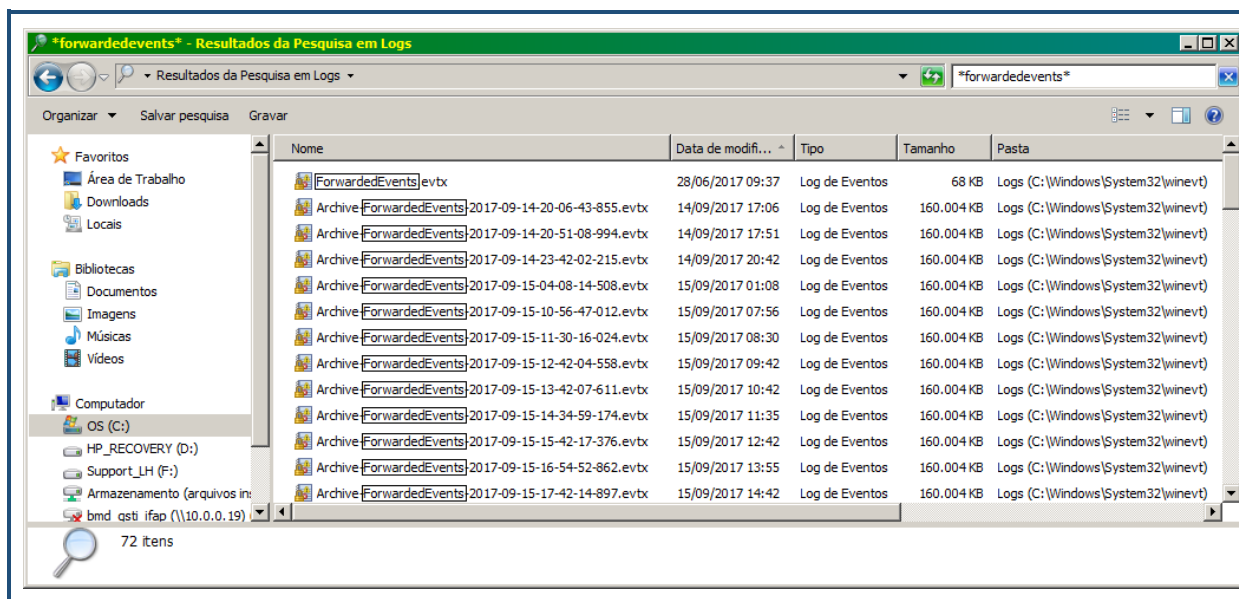
Figura 31: Arquivos rotacionados dos registros de eventos, no servidor origem



Fonte: Cenário real do IFAP – próprio autor

Vale ressaltar que os *logs* coletados, pela estação coletora: RTDTIPC012725, são registros de *logs* baseados em filtros realizados sobre os *logs* gerados pelo servidor de origem: SVR-AD01. E o armazenamento destes arquivos, na estação coletora, é realizado conforme o tamanho definido (150 *Megabytes*) para o rotacionamento dos arquivos e são armazenados no caminho: %SystemRoot%\System32\Winevt\Logs\, como é mostrado na Figura 32.

Figura 32: Arquivos rotacionados dos registros de eventos, na estação coletora



Fonte: Cenário real do IFAP – próprio autor

5.2 Aplicando a Técnica Adotada para a Análise e Auditoria de Logs

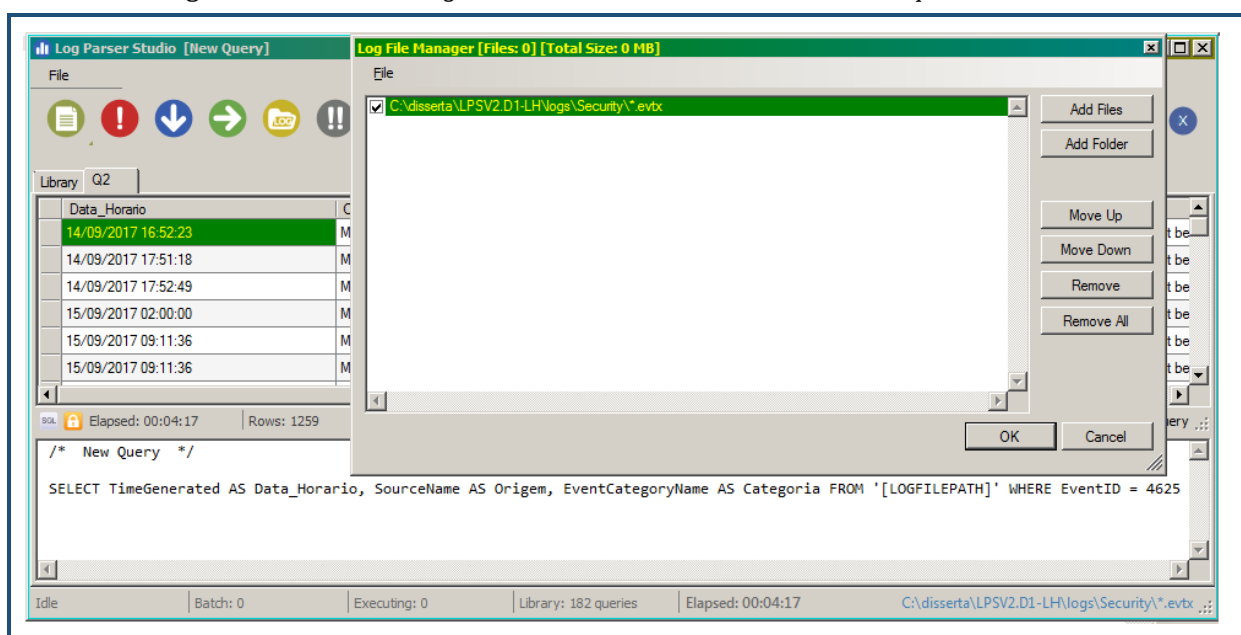
Utilizando os arquivos armazenados pela estação coletora de logs, que são arquivos já fechados pelo processo de rotacionamento, como mostrado na Figura 32 e, com o uso das ferramentas *Microsoft Log Parser 2.2*¹⁷ em ambiente de linha de comandos e a ferramenta *Microsoft Log Parser Studio 2.0 (LPS)*¹⁸, em ambiente gráfico, insere-se as *queries* SQL, específicas para a extração das informações do arquivo de Eventos de Segurança, detalhando que os eventos que estão em análise nessa pesquisa são os elencados na Tabela 10. Ambas as ferramentas se propõem a funcionar como uma ferramenta de suporte ao Administrador de Redes, com o fim de realizar consultas aos arquivos de registros. Na comparação dos resultados fornecidos pelas ferramentas *Log Parse 2.2* e *Log Parser Studio 2.0*, adotou-se a ferramenta LPS 2.0 pois esta ferramenta apresenta recursos e facilidades em relação à ferramenta LP 2.2, pois a ferramenta de linha de comandos não permite a construção de *scripts* PS1, que são *scripts* multiplataforma *Windows*, uma vez que somente possui o recurso de digitação das sintaxes em linha de comando. As possibilidades de consultas utilizando a ferramenta LP 2.2, não estão limitadas aos tipos e qualidade dos relatórios gerados no ambiente de linha de comandos, pois é capaz de gerar saída em formato CSV, mas estão limitadas aos recursos disponíveis para a exportação das *queries* SQL, para os formatos que permitam a interação com outros ambientes *Windows*, a exemplo dos *scripts* PS1, que não necessitam da instalação de ferramentas adicionais.

¹⁷ Download em 27/06/2017: <https://www.microsoft.com/en-us/download/details.aspx?id=24659>

¹⁸ Download em 27/06/2017: <https://gallery.technet.microsoft.com/Log-Parser-Studio-cd458765>

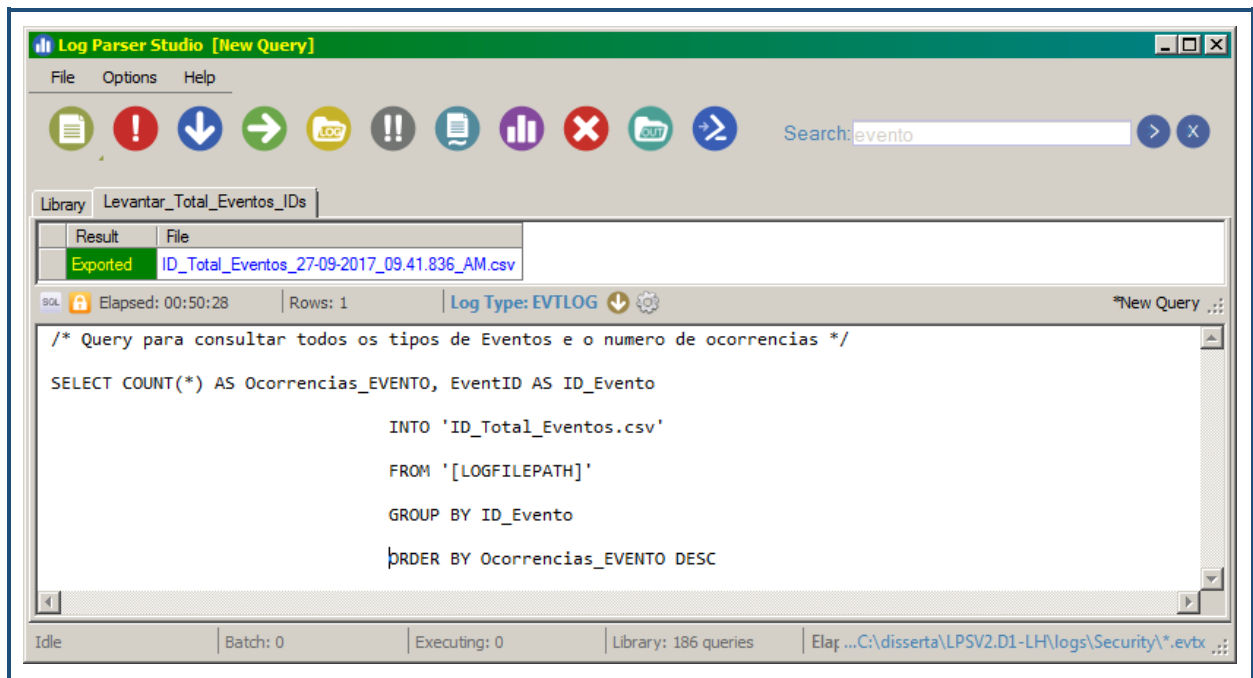
A técnica adotada para a realização de análises nos *logs* de eventos, foi a utilização da ferramenta analisador de *logs*, *Microsoft Log Parser Studio 2.0* que têm como vantagem, além da interface gráfica, a funcionalidade de exportar as *queries* SQL para o formato de *scripts* PS1, o que permite utilizar os *scripts* resultantes em qualquer outro ambiente *Windows*, pelo fato de que o ambiente *PowerShell*, ser componente básico nos sistemas *Windows* e, como enorme diferencial destaca-se que a ferramenta LPS 2.0 possui o recurso de uma biblioteca de *queries* pré-configuradas para consultas específicas, em especial consultas para os sistemas *MS-Exchange* e *MS-Internet Information Service*, outra característica marcante da ferramenta LPS 2.0 é a possibilidade de efetuar a análise da mesma *query* em um lote de arquivos em uma só execução, bastando indicar o diretório onde os arquivos se encontram e, na sintaxe da *query* indicar que a pesquisa é em lote, utilizando o indicador de local da pesquisa: '[LOGFILEPATH]'. E a abordagem de análise aqui proposta é obter as informações do montante, em lote, de arquivos coletados do servidor de origem. E a configuração para usar os arquivos em lotes é mostrada na Figura 33.

Figura 33: Ferramenta *Log Parser Studio 2.0* – análise em lotes de arquivos de eventos



Fonte: Cenário real do IFAP – próprio autor

E definida a ferramenta para a análise de *logs*, defina a forma de abordagem dos arquivos de dados que contém os registros coletados, inicia-se a construção das *queries* SQL voltadas a extrair as informações pertinentes ao Gerenciamento da Segurança no Serviço de Autenticação (*Active Directory*), de acordo com os eventos elencados na Tabela 10. E, antes de iniciar a construção das *queries* e dos *scripts* de interesse, se pesquisou todos os eventos ocorridos em todos os arquivos gerados pela coleta, conforme a consulta SQL mostrada na Figura 34.

Figura 34: Query – todos os eventos ocorridos nos registros coletados

Fonte: Cenário real do IFAP – próprio autor

O tempo de processamento da consulta e geração do resultado no formato CSV, de todos os eventos coletados, durou 50 minutos e 28 segundos. E o resultado da consulta é mostrado na Figura 35.

Figura 35: Relação de todos os eventos ocorridos nos registros coletados

The screenshot shows the LibreOffice Calc spreadsheet titled 'ID_Total_Eventos_26-09-2017_10.57.880_AM.csv'. The spreadsheet has two columns: 'Ocorrencias_EVENTO' and 'ID_Evento'. The data is as follows:

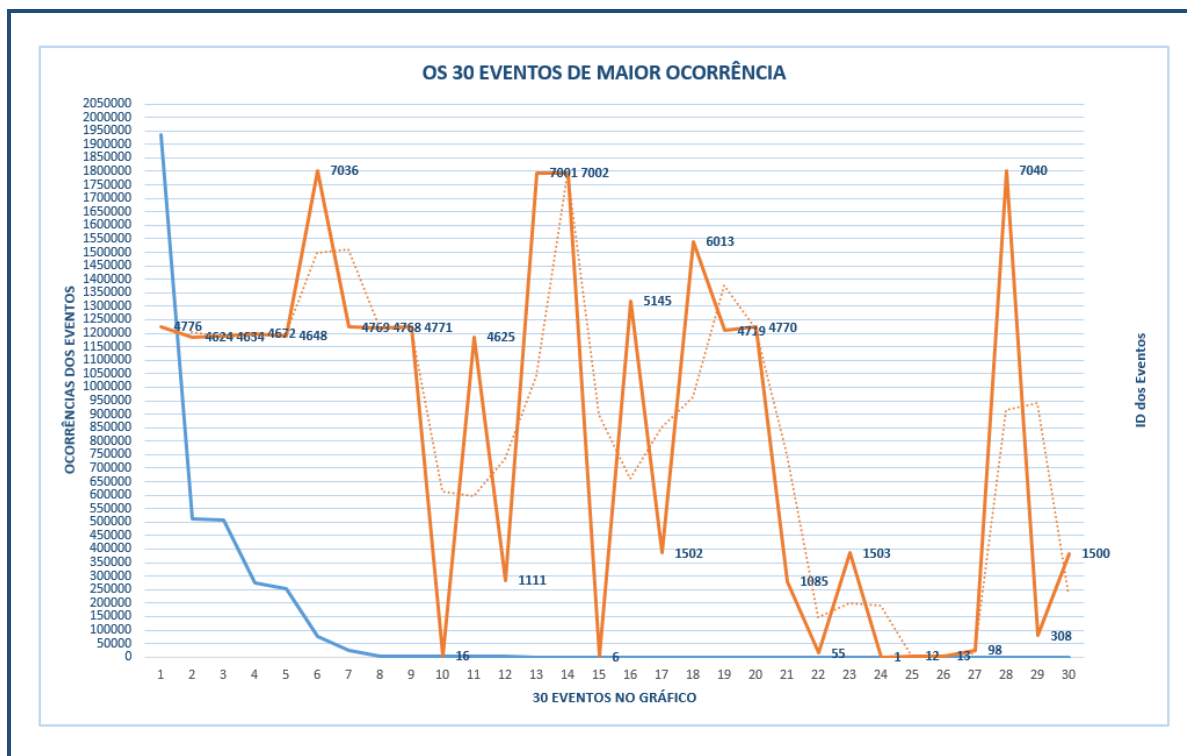
Ocorrencias_EVENTO	ID_Evento
2136857	4776
541893	4624
539305	4634
279190	4672
254420	4648
76918	7036
25699	4769
5824	4768
2919	4771
1976	16
1450	4625
930	1111
568	7001
541	7002
538	6
475	5145
467	1502
448	4719
404	6013
332	4770
251	1085
220	55
209	1503

The status bar at the bottom shows 'Planilha 1 de 1', 'Padrão', 'Média: ; Soma: 0', and '100%'.

Fonte: Cenário real do IFAP – próprio autor

Para demonstrar o universo total das ocorrências dos Eventos de Segurança nos arquivos coletados gerou-se um gráfico, conforme o apresentado na Figura 36.

Figura 36: Gráfico dos tipos de eventos *versus* o número de ocorrências

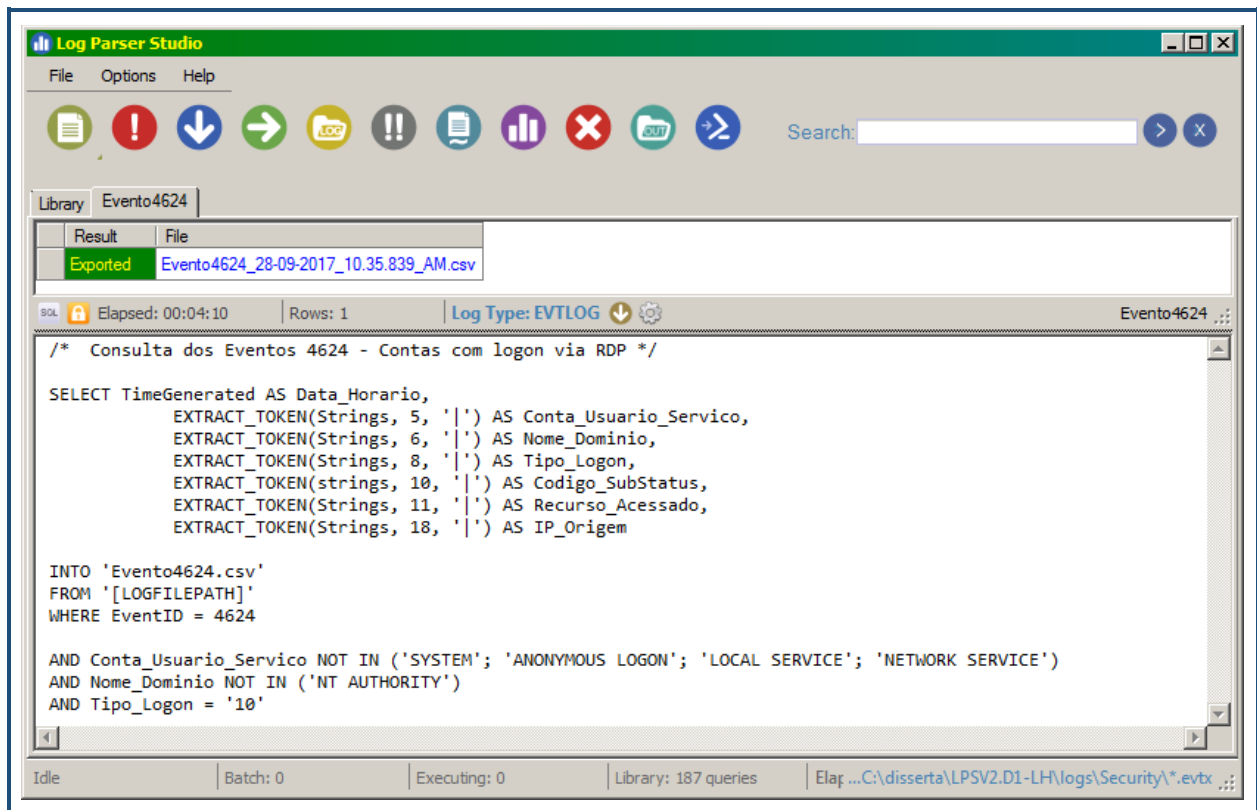


Fonte: Cenário real do IFAP – próprio autor

Os eventos propostos para a análise nesta pesquisa foram definidos conforme a Tabela 10 e, a construção das *queries* e dos *scripts* PS1, é a seguinte:

- a) Listar os eventos de ID 4624, com o objetivo de localizar e identificar as contas de usuários que efetuaram acesso remoto (RDP – *Remote Desktop Connection*), a *query* está construída na Figura 37;

Figura 37: Query 01 – encontrar os eventos 4624 – logons via RDP



Fonte: Cenário real do IFAP – próprio autor

O tempo desprendido no processamento desta consulta e geração da saída em formato CSV, considerando que o volume dos arquivos pesquisados é de 11 *Gigabytes*, foi exatamente de 04 minutos e 10 segundos e, o resultado desta consulta é mostrada na Figura 38.

Figura 38: Arquivo de saída do resultado da consulta aos ID 4624

The screenshot shows the LibreOffice Calc application with the file 'Evento4624_28-09-2017_10.35.839_AM.csv'. The data is displayed in a table with the following columns: A (Data_Horario), B (Conta_Usuario_Servico), C (Nome_Dominio), D (Tipo_Logon), E (Codigo_SubStatus), F (Recurso_Acessado), and G (IP_Origem). The data is as follows:

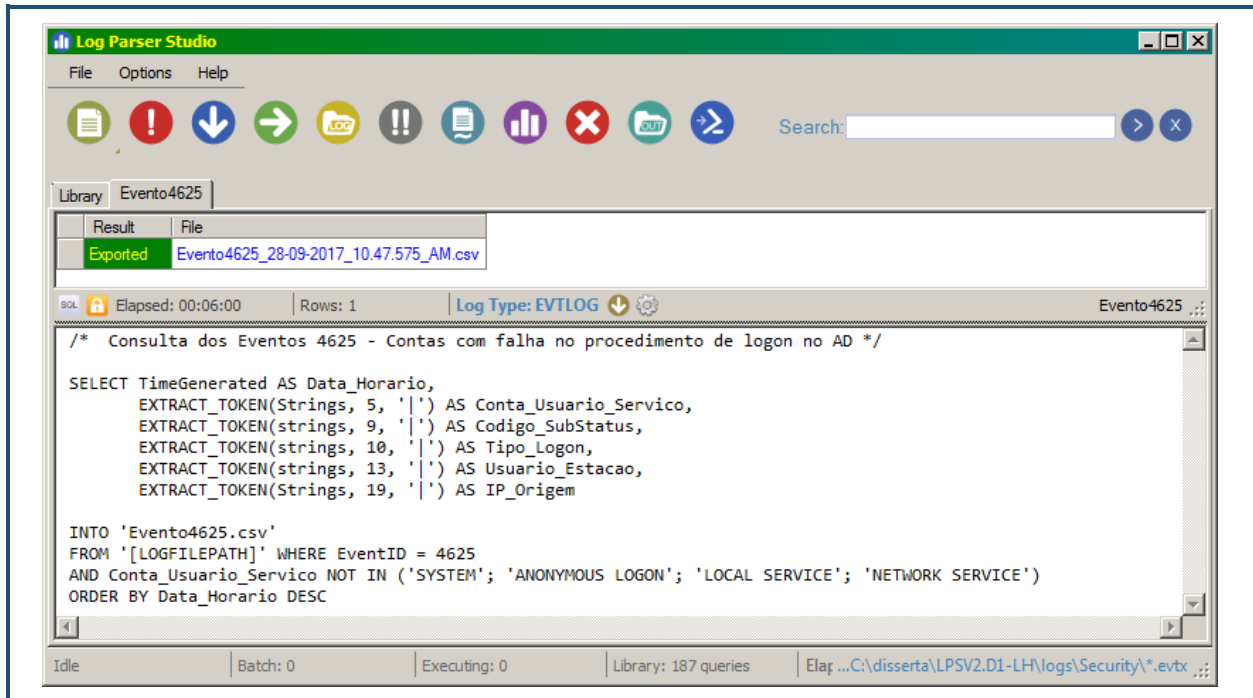
	A	B	C	D	E	F	G
	Data_Horario	Conta_Usuario_Servico	Nome_Dominio	Tipo_Logon	Codigo_SubStatus	Recurso_Acessado	IP_Origem
1	15/09/17 14:27	souza	IFAP	10	Negotiate	SVR-AD01	10.0.0.44
2	15/09/17 14:27	souza	IFAP	10	Negotiate	SVR-AD01	10.0.0.44
3	15/09/17 14:44	souza	IFAP	10	Negotiate	SVR-AD01	10.0.0.44
4	15/09/17 14:44	souza	IFAP	10	Negotiate	SVR-AD01	10.0.0.44
5	15/09/17 16:07	souza	IFAP	10	Negotiate	SVR-AD01	10.0.0.44
6	15/09/17 16:07	souza	IFAP	10	Negotiate	SVR-AD01	10.0.0.44
7	18/09/17 16:29	oliveira	IFAP	10	Negotiate	SVR-AD01	10.0.0.129

The status bar at the bottom shows 'Planilha 1 de 1', 'Padrão', 'Média: 21503,3437037037; Soma: 43006,6874074074', and '100%'.

Fonte: Cenário real do IFAP – próprio autor

- b) Listar os eventos de ID 4625, no servidor de autenticação, com o fim de localizar e identificar usuários com dificuldades de autenticar-se, ou ainda, uma possível tentativa de uso de credenciais alheias, a *query* está construída na Figura 39;

Figura 39: Query 02 – encontrar os eventos 4625 – erros nas tentativas de *logon*



Fonte: Cenário real do IFAP – próprio autor

O tempo desprendido no processamento desta consulta e geração da saída em formato CSV, considerando que o volume dos arquivos pesquisados é de 11 *Gigabytes*, foi exatamente de 06 minutos e 00 segundos e, o resultado desta consulta é mostrada na Figura 40.

Figura 40: Arquivo de saída do resultado da consulta aos ID 4625

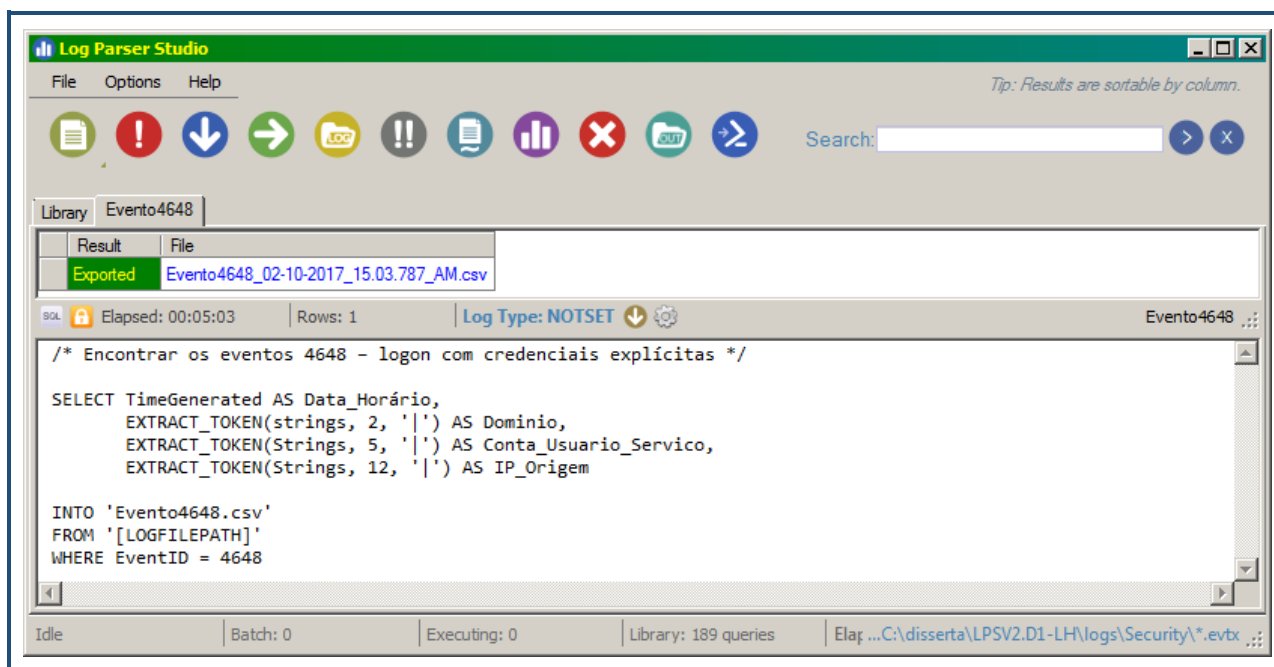
The screenshot shows the LibreOffice Calc application displaying the output of the query. The data is organized into columns: Data_Horario, Conta_Usuario_Servico, Codigo_SubStatus, Tipo_Logon, Usuario_Estacao, and IP_Origem. The table lists several failed login attempts with timestamps and user details.

	A	B	C	D	E	F
	Data_Horario	Conta_Usuario_Servico	Codigo_SubStatus	Tipo_Logon	Usuario_Estacao	IP_Origem
1	14/09/17 16:52	1880933	0xc000006a	3 SVR-AD01		10.0.0.26
2	14/09/17 17:51	impressora	0xc000006a	3 IMP-PROEXT		10.0.0.231
3	14/09/17 17:52	impressora	0xc000006a	3 IMP-PROEXT		10.0.0.231
4	15/09/17 02:00	pantoja	0xc000006a	4 SVR-AD01		-
5	15/09/17 09:11	Suporte IFAP	0xc000006a	3 RTDTIPC000331		10.0.0.209
6	15/09/17 11:13	impressora	0xc000006a	3 IMP-PROEXT		10.0.0.231
7	15/09/17 11:55	gouveia	0xc0000071	3 SVR-AD01		10.0.0.5

Fonte: Cenário real do IFAP – próprio autor

- c) Listar os eventos de ID 4648, com o objetivo de localizar e identificar quais usuários ou serviços estão acessando serviços e recursos utilizando credenciais explícitas, que é quando u, usuário se conecta a um servidor ou executa um programa localmente usando credenciais alternativas, a *query* está construída na Figura 41;

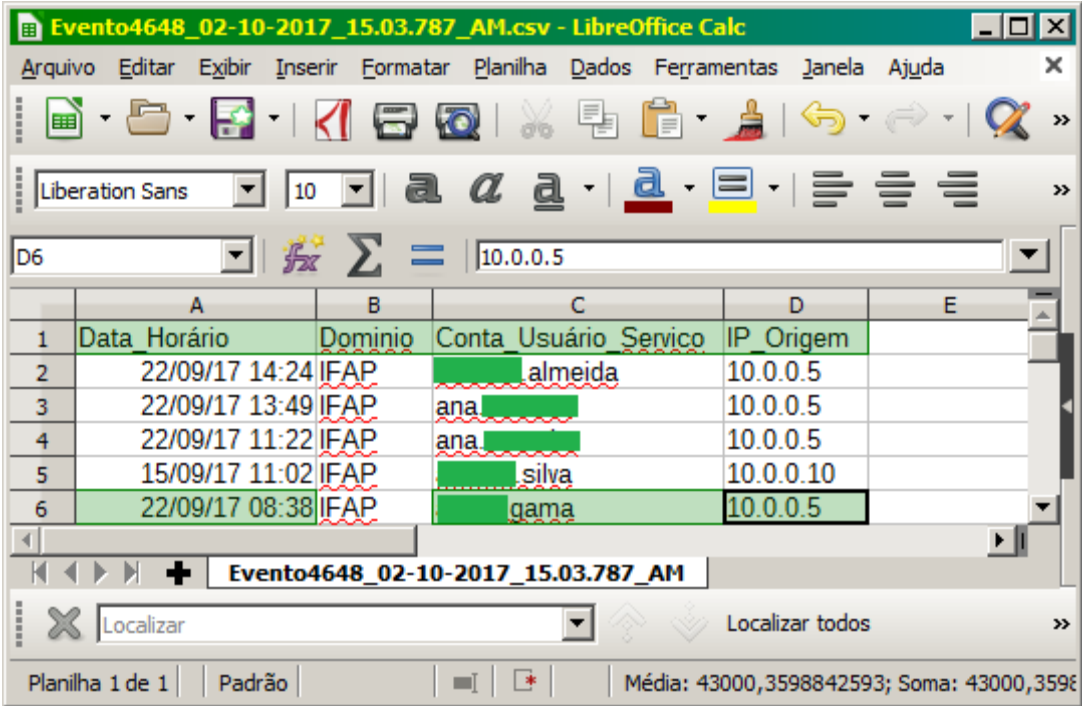
Figura 41: Query 03 – encontrar os eventos 4648 – *logon* com credenciais explícitas



Fonte: Cenário real do IFAP – próprio autor

O tempo despendido no processamento desta consulta e geração da saída em formato CSV, considerando que o volume dos arquivos pesquisados é de 11 *Gigabytes*, foi exatamente de 05 minutos e 03 segundos e, o resultado desta consulta é mostrada na Figura 42.

Figura 42: Arquivo de saída do resultado da consulta aos ID 4648

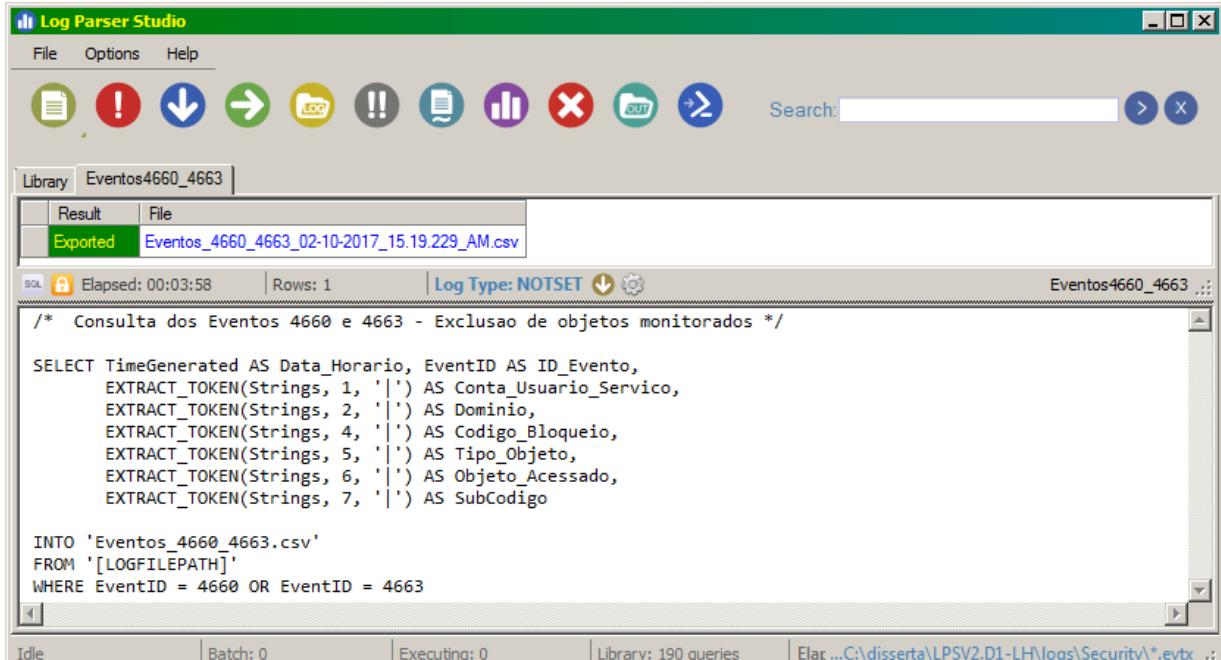


	A	B	C	D	E
1	Data	Horário	Dominio	Conta	Usuário Serviço IP Origem
2	22/09/17	14:24	IFAP	almeida	10.0.0.5
3	22/09/17	13:49	IFAP	ana	10.0.0.5
4	22/09/17	11:22	IFAP	ana	10.0.0.5
5	15/09/17	11:02	IFAP	silva	10.0.0.10
6	22/09/17	08:38	IFAP	gama	10.0.0.5

Fonte: Cenário real do IFAP – próprio autor

- d) Listar os eventos de ID 4660 e 4663, com o objetivo de localizar e identificar objetos excluídos por usuários, quando se está monitorando o compartilhamento de arquivos, a *query* está construída na Figura 43;

Figura 43: Query 04 – encontrar os eventos 4660 e 4663 – exclusão de arquivos



```

/* Consulta dos Eventos 4660 e 4663 - Exclusao de objetos monitorados */

SELECT TimeGenerated AS Data_Horario, EventID AS ID_Evento,
EXTRACT_TOKEN(Strings, 1, '|') AS Conta_Usuario_Servico,
EXTRACT_TOKEN(Strings, 2, '|') AS Dominio,
EXTRACT_TOKEN(Strings, 4, '|') AS Codigo_Bloqueio,
EXTRACT_TOKEN(Strings, 5, '|') AS Tipo_Objeto,
EXTRACT_TOKEN(Strings, 6, '|') AS Objeto_Acessado,
EXTRACT_TOKEN(Strings, 7, '|') AS SubCodigo

INTO 'Eventos_4660_4663.csv'
FROM '[LOGFILEPATH]'
WHERE EventID = 4660 OR EventID = 4663
  
```

Fonte: Cenário real do IFAP – próprio autor

O tempo desprendido no processamento desta consulta e geração da saída em formato CSV, considerando que o volume dos arquivos pesquisados é de 11 *Gigabytes*, foi exatamente de 03 minutos e 58 segundos e, o resultado desta consulta é mostrada na Figura 44.

Figura 44: Arquivo de saída do resultado da consulta aos IDs 4660 e 4663

A	B	C	D	E	F	G
Data_Horario	ID_Evento	Conta_Usuario_Servico	Dominio	Codigo_Bloqueio	Tipo_Objeto	Objeto_Acessado
21/09/17 16:30	4663	costa	IFAP	Security	File	E:\storage_ifap\gabinete\dexp\2017\...
22/09/17 08:33	4663	santos	IFAP	Security	File	E:\storage_ifap\prod\diplag\CERTID...
22/09/17 08:33	4660	santos	IFAP	Security	0x3578	0x4

Fonte: Cenário real do IFAP – próprio autor

- e) Listar os eventos de ID 4771, com o objetivo de localizar e identificar quais usuários, naquele momento, estão com a conta bloqueada após falhas de *logon*, a *query* está construída na Figura 45.

Figura 45: Query 05 – encontrar os eventos 4771 – contas bloqueadas

```

/* Consulta dos Eventos 4771 - Contas bloqueadas */

SELECT TimeGenerated AS Data_Horario,
       EXTRACT_TOKEN(Strings, 0, '|') AS Conta_Usuario_Servico,
       EXTRACT_TOKEN(Strings, 4, '|') AS Codigo_Bloqueio,
       EXTRACT_TOKEN(Strings, 6, '|') AS IP_Estacao

INTO 'Evento4771.csv'
FROM '[LOGFILEPATH]'
WHERE EventID = 4771
  
```

Fonte: Cenário real do IFAP – próprio autor

O tempo desprendido no processamento desta consulta e geração da saída em formato CSV, considerando que o volume dos arquivos pesquisados é de 11 *Gigabytes*, foi exatamente de 04 minutos e 02 segundos e, o resultado desta consulta é mostrada na Figura 46.

Figura 46: Arquivo de saída do resultado da consulta aos IDs 4771

	A	B	C	D
	Data	Horario	Conta	Usuario
1	14/09/17	16:18	almeida	0x18
2	14/09/17	16:41	silva	0x18
3	14/09/17	16:44	moura	0x18
4	14/09/17	16:54	angela	0x18
5	14/09/17	16:54	angela	0x18
6	14/09/17	17:00	batista	0x18
7	14/09/17	17:08	leandro	0x18

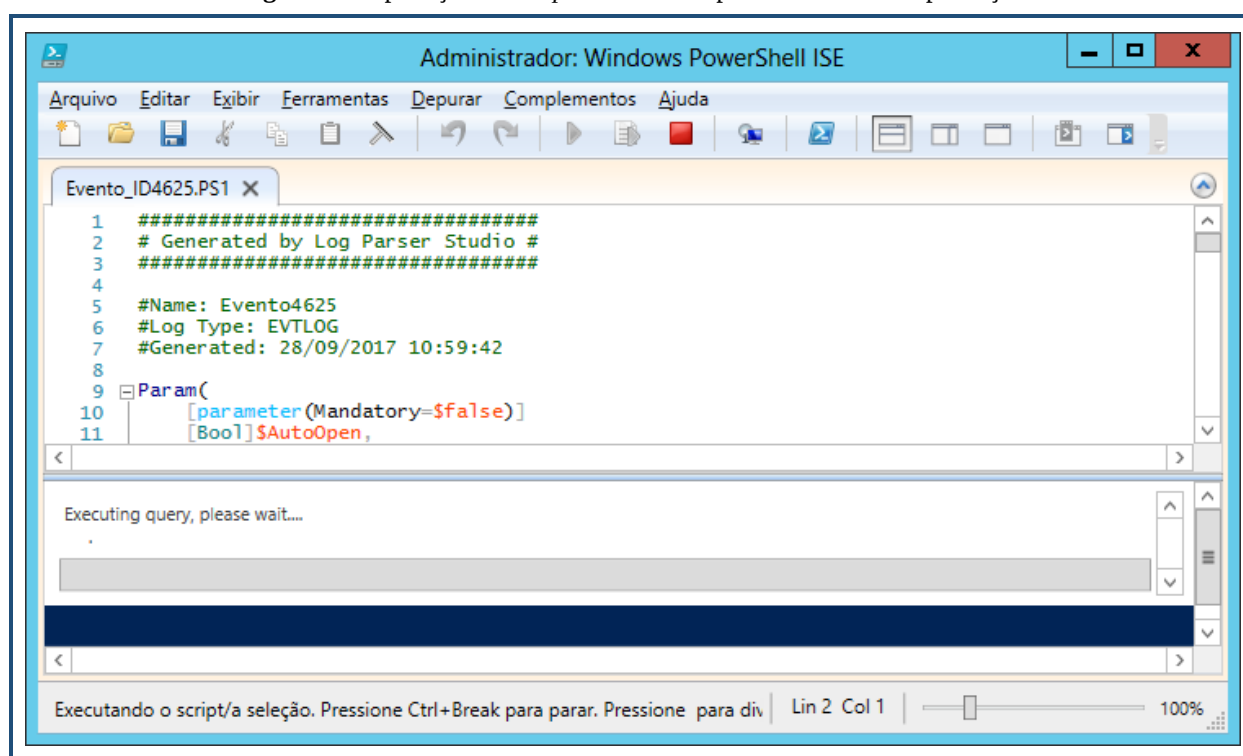
Fonte: Cenário real do IFAP – próprio autor

5.3 Resultados Alcançados

A atividade diária do Administrador de Redes no que se refere à análise dos registros de eventos, com o foco de se identificar as ações praticadas por usuários da rede, torna-se mais fácil e transparente para o processo de auditoria quando se utiliza um ferramental acessível e de manuseio intuitivo e, na utilização de *scripts* próprios para a identificação de eventos específicos gera-se relatórios claros e diretos sobre a informação buscada, reforçando-se que esses relatórios aqui apresentados não estão disponíveis nativamente na plataforma dos sistemas *Windows*. Há uma gama de informações úteis extraídas dos registros de eventos trabalhados nesta pesquisa e, em alinhamento com os modelos de análise aplicados com a ferramenta analisador de *logs*, se conseguiu relatórios contendo informações importantes para a Gerência de Segurança, como data e horário, conta de usuário ou serviço, endereço IP, recurso acessado, detalhamento do evento principal pesquisado. Então esta pesquisa alcançou os seus objetivos propostos, pois conseguiu-se traçar um modelo para o uso de uma ferramenta (LPS 2.0) na construção de outros recursos ferramentais (*scripts* PS1), com o fim de complementar os procedimentos do Gerenciamento de Segurança na rede interna do IFAP.

Para se demonstrar o uso prático do ferramental resultante, observando-se que para a execução em ambiente de produção, modificou-se somente o nome do caminho dos arquivos de pesquisa, pelo nome do instrumento do sistema que gerencia o registro de Eventos de Segurança, alterando-se o *script* gerado com as informações do ambiente de testes, onde a cláusula FROM 'C:\disserta\LPSV2.D1-LH\logs\Security*.evtx', utilizada no arquivo do *script* PS1 a ser aplicado no ambiente de produção, para a cláusula FROM Securiry. Registre-se que ao realizar essa alteração no *script*, a busca efetivada pelo evento só ocorrerá no arquivo *online* que está aberto pelo Servidor de origem e não nos arquivos já rotacionados, pois para tanto basta especificar o caminho de busca dos arquivos que foram fechados e estão no caminho dos logs: C:\Windows\System32\winevt\Logs*.evtx, A aplicação do *script* Evento4625.ps1 no ambiente de produção, no arquivo *online* é demonstrado na Figura 47.

Figura 47: Aplicação do *script* Evento4625.ps1 no servidor em produção



Fonte: Cenário real do IFAP – próprio autor

A execução do *script* no ambiente de produção consumiu o tempo em torno de 12 segundos e, o tamanho do arquivo *online* que foi processado era de 120 *Megabytes*, gerando um arquivo CSV no caminho de saída padrão, que é a pasta Documentos, do perfil que executou o *script* e, o resultado desta consulta ao arquivo *online* é mostrado na Figura 48.

Figura 48: Arquivo de saída da consulta *online* ao Evento *Security* – ID 4625

Evento4625_PRODUCAO.csv - LibreOffice Calc

Arquivo Editar Exibir Inserir Formatar Planilha Dados Ferramentas Janela Ajuda

Fonte: Cenário real do IFAP – próprio autor

O modelo proposto e os exemplos colocados em prática nesta pesquisa, para a intervenção na Gestão de Segurança da Informação dentro das unidades da TIC dos Institutos Federais de Educação, tem como objetivo o monitoramento dos eventos de auditoria e segurança a partir da coleta sistemática de *logs* de eventos, pois a preocupação e a atenção aos registros de eventos, praticamente, não existem dentro dos IFs. E considerando que no conjunto dos serviços de rede implementados e das ferramentas disponibilizadas, com o fim de garantir e monitorar a segurança da informação nos IFs, a retenção e o tratamento de *logs* são as atividades que nem mesmo estão no planejamento da segurança de dados e, são exatamente, os registros de eventos que servirão de trilhas para processos de auditoria de possíveis incidentes ou eventos que venha a comprometer os serviços e recursos computacionais.

6

CONCLUSÃO E TRABALHOS FUTUROS

Este capítulo apresenta as informações acerca das conclusões, as dificuldades encontradas e as contribuições e recomendações para os trabalhos futuros.

6.1 Conclusão

Neste trabalho apresentou-se os conceitos, estruturas e recomendações sobre a retenção dos eventos de sistemas – *logs*, com fim de se implantar um serviço a mais no ambiente de rede, com o uso de uma estação coletora de *logs*, para futura análise dos eventos, com o objetivo de ser a principal ferramenta para a identificação de problemas, comportamentos e ocorrências relevantes no Gerenciamento da segurança. E, tomando como base outros trabalhos sobre coleta, retenção e gerenciamento de *logs* fechou-se a proposta apresentada neste trabalho em criar a infraestrutura de coleta de *logs*, no modelo *loghost* centralizado (*host-based*). Para se provar os conceitos e o modelo escolhido, configurou-se um ambiente, onde um servidor *Active Directory* encaminhou seus *logs* de segurança para uma estação *Windows 7*, neste ambiente constatou-se que há que se dedicar ao planejamento da retenção de *logs*, de forma apurada, pois a escolha de tamanho de arquivos dos *logs*, seu tempo de retenção, os filtros dos serviços e eventos que se deseja reter, são atributos que combinados podem resultar em arquivos enormes e, que irá impactar na performance de análise e processamento dos dados coletados e, como o comprovado por esta pesquisa, o uso das ferramentas analisadores de *logs* baseadas em ambiente gráfico (GUI), possibilitou o ganho de tempo e de melhores respostas, possibilitando inclusive a geração de estatísticas e gráficos do universo de eventos coletados.

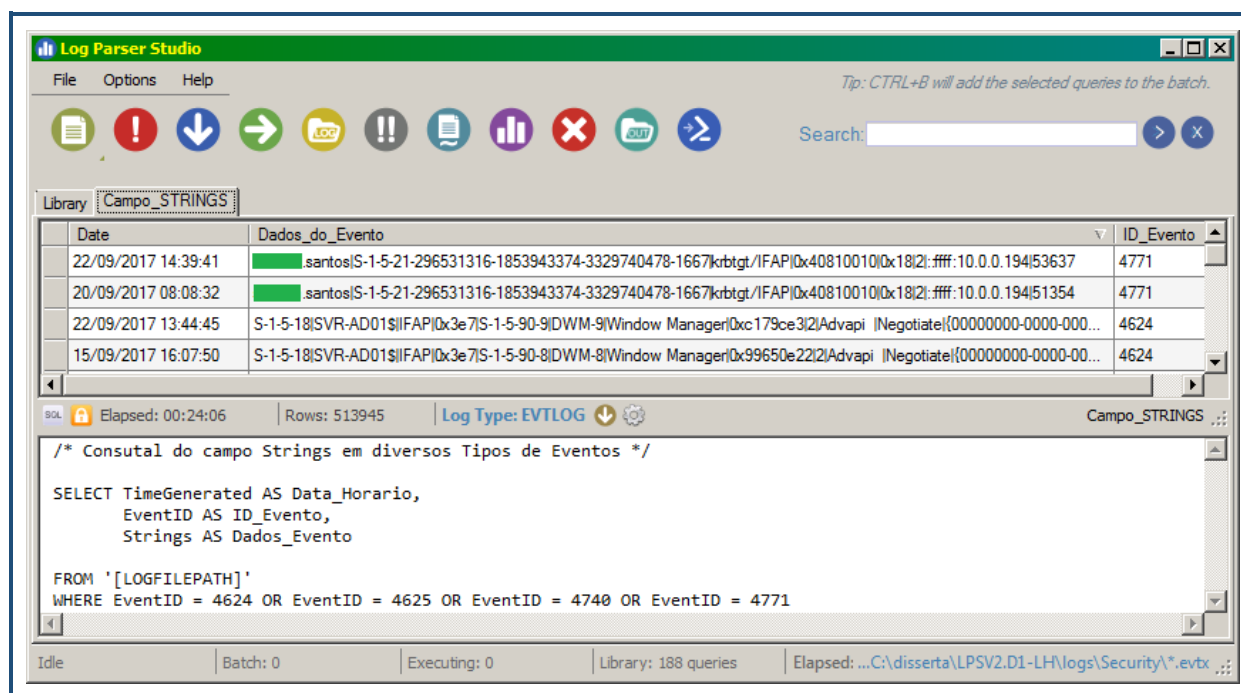
Esta pesquisa se coaduna ao gerenciamento e correlação de eventos de segurança (SIEM), quando demonstra claramente o atendimento aos preceitos do Gerenciamento de Eventos e, permite a inferência de correlação de eventos de segurança ao ofertar ao Administrador de rede o uso de *scripts* .PS1 para a localização e identificação de Eventos de interesse e, ainda, gera relatórios para as consultas efetuadas. O ferramental entregue por esta pesquisa, possibilita a interação *online* com os dados dos Eventos de Segurança, os quais estejam ocorrendo no Serviço Eventos de Segurança – *Security* e, também possibilita a interação com os arquivos de *logs* já

armazenados. Conclui-se nesta pesquisa que as configurações de consultas aos arquivos de eventos nos sistemas *Windows* utilizando *scripts* PS1 são escaláveis, na medida em que os *scripts* gerados podem ser aplicados em ambientes de pequeno e médio porte e, até mesmo em ambientes de grande porte, pois as características e variáveis que mudam entre esses ambientes é o volume de arquivos a serem tratados e, em versões mais antigas do sistema *Windows*, mudam os caminhos de armazenamento dos arquivos de eventos.

6.2 Dificuldades Encontradas

Como exemplo de barreiras enfrentadas para o desenvolvimento desta pesquisa pode-se registrar a falta de fontes de consulta onde constem, com exatidão e suficientes informações, como compor as sintaxes SQL aplicadas na ferramenta *Log Parse Studio 2.0* ou mesmo na ferramenta de linha de comando da ferramenta *Log Parser 2.2*, reforçando que ambas ferramentas são mantidas pela empresa *Microsoft* e, a documentação oficial disponível nos diversos portais de suporte, não reúne informações suficientes para o suporte de uso das ferramentas. Outra barreira encontrada ao longo do desenvolvimento desta pesquisa foi a falta de informações precisas sobre a delimitação dos dados constantes no campo *Strings* dos arquivos EVTX dos eventos do sistema *Windows*, pois este campo específico é onde estão reunidas, em um único conjunto de dados, a maioria das informações úteis sobre os eventos ocorridos e, para se encontrar exatamente as informações que se busca, exige-se a realização de testes de extração das informações do campo *Strings* utilizando as delimitações (|) da cláusula `EXTACT_TOKEN`, por várias tentativas até se conseguir a informação buscada e, ainda, este campo *Strings* não mantém um padrão de localização das informações para os tipos de eventos diferentes e, também, armazena muito ruído. Então para cada tipo de pesquisa de eventos, necessita-se descobrir, no campo *Strings*, exatamente a posição da informação buscada e, para ilustrar o que acontece nas extrações de informações do campo *Strings* de um arquivo EVTX, aplicou-se uma consulta SQL para demonstrar o conteúdo completo de deste campo em diferentes eventos, demonstrando-se as diferentes formatações deste campo, como o mostrado na Figura 49.

Figura 49: Informações constantes no campo *Strings* de um arquivo EVT



Fonte: Cenário real do IFAP – próprio autor

Em relação às limitações da abordagem desta pesquisa, pode-se registrar que nesse trabalho não foi abordada a estrutura e nem a análise dos registros de eventos de outras fontes de *logs* no ambiente do sistema *Windows*, a exemplo das fontes padrão de eventos: Aplicativos; Instalação; Sistema, dentre outras fontes de eventos do *Windows* mais especializadas, que são caracterizados conforme a heterogeneidade dos ambientes configurados.

6.3 Trabalhos Futuros

No desenho de um cenário ideal, os Administradores de Rede poderão lançar mão de um modelo de previsão de comportamento de usuários, de forma proativa, o que possibilitaria a identificação de possíveis atacantes ou mesmo descobrindo como melhorar a disponibilização dos serviços de rede. Então, neste sentido, um trabalho futuro de interesse é a aplicação da técnica de extração do conhecimento, através de modelo de descoberta de conhecimentos em bases de dados (KDD – *Knowledge Discovery in Databases*). Há um trabalho de pesquisa recente (PAULAUSKAS; AUSKALNIS, 2017), que trata parte do problema da análise de conjunto de dados, com o fim de levantar possíveis ataques às redes, esta pesquisa aborda a necessidade da preparação dos dados corretamente para a fase de processamento, o que garante resultados precisos e confiáveis na fase de análise dos dados. Inclusive a concentração da pesquisa é em relação à detecção de atacantes ao ambiente computacional, usando como técnica de pré-processamento dos dados, o modelo de descoberta de conhecimentos em bases de dados (KDD).

REFERÊNCIAS

ABNT – Associação Brasileira de Normas Técnicas. **NBR ISO/ IEC 27002:2013: Tecnologia da informação: Técnicas de segurança: Código de prática para controles de segurança da informação**. Rio de Janeiro-RJ, ABNT, 2013.

APACHE.ORG, Log4j. **Logging Services: Log4j 2.8.2 – Architecture**, [Online; accessed 11-July-2017],

<https://logging.apache.org/log4j/2.0/manual/architecture.html>.

CARDO, Nicholas P. **LOGJAM: A Scalable Unified Log File Archiver**, International Conference for High Performance Computing, Networking, Storage and Analysis (SC 2011), Seattle, WA, USA, 2011.

CHENG, Feng., *et al.* **Security Event Correlation supported by Multi-Core Architecture**, 3rd International Conference on IT Convergence and Security (ICITCS 2013), Macao, China, 2013.

CHUVAKIN, Anton. **The Complete Guide to Log and Event Management**, NetIQ, 2016, [Online; accessed 07-April-2017], <https://www.netiq.com/promo/security/the-complete-guide-to-log-and-event-management.html>.

DU, Min.; LI, Feifei. **Spell: Streaming Parsing of System Event Logs**, IEEE 16th International Conference on Data Mining (ICDM 2016), Barcelona, Spain, 2016.

FRASER, Barbara., *et al.* Site Security Handbook. **RFC 2196**, 1997, [Online; accessed 09-April-2017], <https://tools.ietf.org/html/rfc2196>.

GERHARDS, Rainer. The Syslog Protocol. **RFC 5424**, 2009, [Online; accessed 09-April-2017], <https://tools.ietf.org/html/rfc5424>.

GIL, Antonio Carlos. **Como Elaborar Projetos de Pesquisa**. 4th.ed., Atlas, São Paulo, 2002.

GHOSHAL, Devarshi.; PLALE, Beth. **Provenance from Log Files: a BigData Problem**, EDBT/ICDT 2013 Joint Conference, Workshops Pages 290-297, ACM Digital Library, Genoa, Italy, 2013.

KENT, Karen.; SOUPPAYA Murugiah. **Special Publication 800-92: Guide to Computer Security Log Management**, NIST – National Institute of Standards and Technology, Gaithersburg, MD, USA, 2006.

LIM, Chinghway.; SINGH, Navjot.; YAJNIK, Shalini. **A Log Mining Approach to Failure Analysis of Enterprise Telephony Systems**, IEEE International Conference on Dependable Systems and Networks (DSN 2008), Anchorage, Alaska, 2008.

MICROSOFT, Developer Network. **EVENTLOGRECORD structure**, [Online; accessed 10-September-2017], [https://msdn.microsoft.com/pt-br/library/windows/desktop/aa363646\(v=vs.85\).aspx](https://msdn.microsoft.com/pt-br/library/windows/desktop/aa363646(v=vs.85).aspx).

MICROSOFT, Support. **Description of security events in Windows 7 and in Windows Server 2008 R2**, [Online; accessed 11-September-2017],

<https://support.microsoft.com/en-us/help/977519/description-of-security-events-in-windows-7-and-in-windows-server-2008>.

MICROSOFT, TechNet Blogs. **Introducing: Log Parser Studio**, [Online; accessed 20-April-2017],

<https://blogs.technet.microsoft.com/exchange/2012/03/07/introducing-log-parser-studio/>.

MICROSOFT, TechNet Blogs. **LogParser and Powershell - LOGPOWER**, [Online; accessed 11-September-2017], <https://blogs.technet.microsoft.com/sooraj-sec/2017/02/20/logparser-and-powershell-logpower/>.

MICROSOFT, TechNet Blogs. **Logparser play of a forensicator**, [Online; accessed 11-September-2017], <https://blogs.technet.microsoft.com/sooraj-sec/2016/08/20/logparser-play-of-a-forensicator/>.

MICROSOFT, TechNet Library. **Apêndice I: Eventos a serem monitorados**, [Online; accessed 21-July-2017], [https://technet.microsoft.com/pt-br/library/dn535498\(v=ws.11\).aspx](https://technet.microsoft.com/pt-br/library/dn535498(v=ws.11).aspx).

MICROSOFT, TechNet Library. **Configurar computadores para encaminhar e coletar eventos**, [Online; accessed 10-April-2017], [https://technet.microsoft.com/pt-br/library/cc748890\(v=ws.11\).aspx](https://technet.microsoft.com/pt-br/library/cc748890(v=ws.11).aspx).

MICROSOFT, TechNet Library. **Criar uma nova inscrição**, [Online; accessed 10-April-2017], [https://technet.microsoft.com/pt-br/library/cc722010\(v=ws.11\).aspx](https://technet.microsoft.com/pt-br/library/cc722010(v=ws.11).aspx).

MICROSOFT, TechNet Library. **Funcionalidade da Detecção de Intrusão**, [Online; accessed 10-April-2017], <https://technet.microsoft.com/pt-br/library/dd459005.aspx>.

MICROSOFT, TechNet Learn. **Script Center**, [Online; accessed 17-June-2017],

<https://technet.microsoft.com/en-us/scriptcenter/dd919274.aspx>

MYERS, Justin; GRIMAILA, Michael R.; MILLS, Robert F. **Log-Based Distributed Security Event Detection Using Simple Event Correlator**, 44th Hawaii International Conference on System Sciences (HICSS 2011), Kauai, HI, USA, 2011.

NEW, Darren.; ROSE, Marshall T. Reliable Delivery for Syslog. **RFC 3195**, 2001, [Online; accessed 09-April-2017], <https://tools.ietf.org/html/rfc3195>.

NIC.BR, Security Office. **Práticas de Segurança para Administradores de Redes Internet**, [Online; accessed 12-April-2017], <https://www.cert.br/docs/seg-adm-redes/seg-adm-redes.html>.

PAULAUSKAS, Nerijus.; AUSKALNIS, Juozas. **Analysis of Data Pre-processing Influence on Intrusion Detection using NSL-KDD Dataset**, Open Conference of Electrical, Electronic and Information Sciences (eStream 2017), Vilnius, Lithuania, 2017.

PwC, Price waterhouse Coopers. **18ª Pesquisa Global de Segurança da Informação**, [Online; accessed 03-April-2017],

<http://www.pwc.com.br/pt/publicacoes/servicos/assets/consultoria-negocios/2016/tl-gsiss16-pt.pdf>.

SMITH, Russell. **Configure Event Log Forwarding in Windows Server 2012**, Petri IT Knowledgebase, 2015, [Online; accessed 12-April-2017],
<https://www.petri.com/configure-event-log-forwarding-windows-server-2012-r2>.

SON, HS., *et al.* **Network Traffic and Security Event Collecting System**, 2nd International Conference on Electrical Systems, Technology and Information (ICESTI 2015), Bali-Indonesia, 2015, p. 439-446, [Online; accessed 07-April-2017],
https://link.springer.com/chapter/10.1007/978-981-287-988-2_48.

SONG, Jinliang.; LUO, Tiejian.; CHEN, Su. **Behavior Pattern Mining: Apply Process Mining Technology to Common Event Logs of Information Systems**, IEEE International Conference on Networking, Sensing and Control (ICNSC 2008), Sanya, China, 2008.

SONG, Wei., *et al.* **Efficient Alignment Between Event Logs and Process Models**, IEEE Transactions on Services Computing (Volume: 10, Issue: 1), [S.l.], 2017, p. 136-149.

STEARLEY, Jon. **Towards Informatic Analysis of Syslogs**, IEEE International Conference on Cluster Computing (Cluster 2004), San Diego, CA, USA, 2004.

TOMONO, Akihiro., *et al.* **A Log Management System for Internal Control**, 12th International Conference on Network-Based Information Systems (NBIS 2009), Indianapolis, USA, 2009.

TSUNODA, Hiroshi., *et al.* **A Prioritized Retransmission Mechanism for Reliable and Efficient Delivery of Syslog Messages**, IEEE 7th Annual Communication Networks and Services Research Conference (CNSR 2009), Moncton, BC, Canada, 2009.

TSUNODA, Hiroshi.; KEENI, Glenn M. **Managing syslog**, IEEE 16th Asia-Pacific Network Operations and Management Symposium (APNOMS 2014), Hsinchu, Taiwan, 2014.

VAARANDI, Risto.; PIHELAS, Mauno. **Using Security Logs for Collecting and Reporting Technical Security Metrics**, IEEE Military Communications Conference (MILCOM 2014), Baltimore, USA, 2014.

YUE, Tian.; XIAOBIN, Li.; ZHENGQIU, Yang. **The Research And Design Of Log Management System Based On Struts Frame**, International Symposium on Computer Science and Computational Technology, (ISCST 2008), Shanghai, China, 2008.

ZHAOJUN, Gu.; YONG, Li.; WENJING, Niu. **Analysis and Implement of PIX Firewall Syslog Log**, The 2nd IEEE International Conference on Information Management and Engineering (ICIME 2010), Chengdu, China, 2010.

ZHUGE, Chen.; VAARANDI, Risto. **Efficient Event Log Mining with LogClusterC**, IEEE International Conference on Big Data Security on Cloud (BigDataSecurity 2017) Beijing, China, 2017.

APÊNDICES

A

Query – Todos os Eventos Ocorridos nos Registros Coletados

```
/* Query para consultar todos os tipos de Eventos e o numero de ocorrencias */  
SELECT COUNT(*) AS Ocorrencias_EVENTO, EventID AS ID_Evento  
        INTO 'ID_Total_Eventos.csv'  
FROM '[LOGFILEPATH]'  
        GROUP BY ID_Evento  
        ORDER BY Ocorrencias_EVENTO DESC
```

B

Query 01 – Encontrar os Eventos 4624 – Logons Via RDP

```
/* Consulta dos Eventos 4624 - Contas com logon via RDP */  
  
SELECT TimeGenerated AS Data_Horario,  
       EXTRACT_TOKEN(Strings, 5, '|') AS Conta_Usuario_Servico,  
       EXTRACT_TOKEN(Strings, 6, '|') AS Nome_Dominio,  
       EXTRACT_TOKEN(Strings, 8, '|') AS Tipo_Logon,  
       EXTRACT_TOKEN(Strings, 10, '|') ASCodigo_SubStatus,  
       EXTRACT_TOKEN(Strings, 11, '|') AS Recurso_Acessado,  
       EXTRACT_TOKEN(Strings, 18, '|') AS IP_Origem  
  
INTO 'Evento4624.csv'  
FROM '[LOGFILEPATH]'  
WHERE EventID = 4624  
  
AND Conta_Usuario_Servico NOT IN ('SYSTEM'; 'ANONYMOUS LOGON'; 'LOCAL SERVICE';  
 'NETWORK SERVICE')  
AND Nome_Dominio NOT IN ('NT AUTHORITY')  
AND Tipo_Logon = '10'
```

C

Script PS1 – Encontrar os Eventos 4624

```
#####
# Generated by Log Parser Studio #
#####

#Name: Evento4624
#Log Type: EVTLOG
#Generated: 28/09/2017 10:46:25

Param(
    [parameter(Mandatory=$false)]
    [Bool] $AutoOpen,
    [parameter(Mandatory=$false)]
    [String] $OutFile,
    [parameter(Mandatory=$false)]
    [Bool] $IgnoreInParams,
    [parameter(Mandatory=$false)]
    [Bool] $IgnoreOutParams)

$Error.Clear()
$DefaultFolder=[Environment]::GetFolderPath("MyDocuments")
$Destination = "Evento4624.csv"
$Destination = $DefaultFolder + "\" + $Destination

if($OutFile -ne [String]::Empty)
{
    $OutFileType = [System.IO.Path]::GetExtension($OutFile.ToUpper())
    $OriginalFileType = [System.IO.Path]::GetExtension($Destination.ToUpper())
    if($OutFileType -ne $OriginalFileType)
    {
        write-Host "You have chosen" $OutFileType "as the output, but this
script was originally generated as" $OriginalFileType -ForegroundColor Red
        write-Host "Either change -OutFile to" $OriginalFileType "or generate
the script again with the output as" $OutFileType -ForegroundColor Red
        write-Host "You can also modify the OutputFormat variable in this
script to match the correct Log Parser 2.2 COM output format." -ForegroundColor Red
        [System.Environment]::NewLine
        return
    }
    else
    {
        if($true -ne $OutFile.Contains("\"))
        {
            $Destination = $DefaultFolder + "\" + $OutFile
        }
        else
        {
            $Destination = $OutFile
        }
    }
}

$LogQuery = New-Object -ComObject "MSUtil.LogQuery"
$InputFormat = New-Object -ComObject "MSUtil.LogQuery.EventLogInputFormat"
$OutputFormat = New-Object -ComObject "MSUtil.LogQuery.CSVOutputFormat"

if($IgnoreInParams-eq $false){
    $InputFormat.fullText=1
    $InputFormat.resolveSIDs=0
    $InputFormat.formatMsg=1
    $InputFormat.msgErrorMode="MSG"
    $InputFormat.fullEventCode=0
    $InputFormat.direction="FW"
```

```

        $InputFormat.stringsSep="|"
        $InputFormat.binaryFormat="PRINT"
        $InputFormat.ignoreMessageErrors=1
    }
    if($IgnoreOutParams -eq $false){
        $OutputFormat.Headers="AUTO"
        $OutputFormat.oDQuotes="AUTO"
        $OutputFormat.tabs=0
        $OutputFormat.oTsFormat="yyyy-MM-dd hh:mm:ss"
        $OutputFormat.oCodepage=0
        $OutputFormat.fileMode=1
    }
    Write-Progress -Activity "Executing query, please wait..." -Status " "

    $SQLQuery = "SELECT TimeGenerated AS Data_Horario, EXTRACT_TOKEN(Strings, 5, '|') AS
Conta_Usuario_Servico, EXTRACT_TOKEN(Strings, 6, '|') AS Nome_Dominio,
EXTRACT_TOKEN(Strings, 8, '|') AS Tipo_Logon, EXTRACT_TOKEN(strings, 10, '|') AS
Codigo_SubStatus, EXTRACT_TOKEN(Strings, 11, '|') AS Recurso_Acessado,
EXTRACT_TOKEN(Strings, 18, '|') AS IP_Origem INTO '"' + $Destination + '"' FROM
'C:\disserta\LPSV2.D1-LH\logs\Security\*.evtx' WHERE EventID = 4624 AND
Conta_Usuario_Servico NOT IN ('SYSTEM'; 'ANONYMOUS LOGON'; 'LOCAL SERVICE'; 'NETWORK
SERVICE') AND Nome_Dominio NOT IN ('NT AUTHORITY') AND Tipo_Logon = '10'"
    $rtnVal = $LogQuery.ExecuteBatch($SQLQuery, $InputFormat, $OutputFormat);
    $OutputFormat = $null;
    $InputFormat = $null;
    $LogQuery = $null;

    if($AutoOpen)
    {
        try
        {
            Start-Process($Destination)
        }
        catch
        {
            Write-Host $_.Exception.Message -ForegroundColor Red
            Write-Host $_.Exception.GetType().FullName -ForegroundColor Red
            Write-Host "NOTE: No output file will be created if the query returned
zero records!" -ForegroundColor Gray
        }
    }
}

```

D

Query 02 – Encontrar os Eventos 4625 – Erros nas Tentativas de Logon

```
/* Consulta dos Eventos 4625 - Contas com falha no procedimento de logon no AD */  
  
SELECT TimeGenerated AS Data_Horario,  
       EXTRACT_TOKEN(Strings, 5, '|') AS Conta_Usuario_Servico,  
       EXTRACT_TOKEN(strings, 9, '|') ASCodigo_SubStatus,  
       EXTRACT_TOKEN(strings, 10, '|') AS Tipo_Logon,  
       EXTRACT_TOKEN(strings, 13, '|') AS Usuario_Estacao,  
       EXTRACT_TOKEN(Strings, 19, '|') AS IP_Origem  
  
INTO 'Evento4625.csv'  
FROM '[LOGFILEPATH]' WHERE EventID = 4625  
AND Conta_Usuario_Servico NOT IN ('SYSTEM'; 'ANONYMOUS LOGON'; 'LOCAL SERVICE';  
'NETWORK SERVICE')  
ORDER BY Data_Horario DESC
```

E

Script PS1 – Encontrar os Eventos 4625

```
#####
# Generated by Log Parser Studio #
#####

#Name: Evento4625
#Log Type: EVTLOG
#Generated: 28/09/2017 10:59:42

Param(
    [parameter(Mandatory=$false)]
    [Bool] $AutoOpen,
    [parameter(Mandatory=$false)]
    [String] $OutFile,
    [parameter(Mandatory=$false)]
    [Bool] $IgnoreInParams,
    [parameter(Mandatory=$false)]
    [Bool] $IgnoreOutParams)

$Error.Clear()
$DefaultFolder=[Environment]::GetFolderPath("MyDocuments")
$Destination = "Evento4625.csv"
$Destination = $DefaultFolder + "\" + $Destination

if($OutFile -ne [String]::Empty)
{
    $OutFileType = [System.IO.Path]::GetExtension($OutFile.ToUpper())
    $OriginalFileType = [System.IO.Path]::GetExtension($Destination.ToUpper())
    if($OutFileType -ne $OriginalFileType)
    {
        Write-Host "You have chosen" $OutFileType "as the output, but this
script was originally generated as" $OriginalFileType -ForegroundColor Red
        Write-Host "Either change -OutFile to" $OriginalFileType "or generate
the script again with the output as" $OutFileType -ForegroundColor Red
        Write-Host "You can also modify the OutputFormat variable in this
script to match the correct Log Parser 2.2 COM output format." -ForegroundColor Red
        [System.Environment]::NewLine
        return
    }
    else
    {
        if($true -ne $OutFile.Contains("\"))
        {
            $Destination = $DefaultFolder + "\" + $OutFile
        }
        else
        {
            $Destination = $OutFile
        }
    }
}

$LogQuery = New-Object -ComObject "MSUtil.LogQuery"
$InputFormat = New-Object -ComObject "MSUtil.LogQuery.EventLogInputFormat"
$OutputFormat = New-Object -ComObject "MSUtil.LogQuery.CSVOutputFormat"

if($IgnoreInParams-eq $false){
    $InputFormat.fullText=1
    $InputFormat.resolveSIDs=0
    $InputFormat.formatMsg=1
    $InputFormat.msgErrorMode="MSG"
    $InputFormat.fullEventCode=0
    $InputFormat.direction="Fw"
    $InputFormat.stringsSep="|"
    $InputFormat.binaryFormat="PRINT"
}
```

```

        $InputFormat.ignoreMessageErrors=1
    }
    if($IgnoreOutParams -eq $false){
        $OutputFormat.Headers="AUTO"
        $OutputFormat.oDQuotes="AUTO"
        $OutputFormat.tabs=0
        $OutputFormat.oTsFormat="yyyy-MM-dd hh:mm:ss"
        $OutputFormat.oCodepage=0
        $OutputFormat.fileMode=1
    }
    Write-Progress -Activity "Executing query, please wait..." -Status " "

    $SQLQuery = "SELECT TimeGenerated AS Data_Horario, EXTRACT_TOKEN(Strings, 5, '|')
    AS Conta_Usuario_Servico, EXTRACT_TOKEN(strings, 9, '|') AS codigo_SubStatus,
    EXTRACT_TOKEN(strings, 10, '|') AS Tipo_Logon, EXTRACT_TOKEN(strings, 13, '|') AS
    Usuario_Estacao, EXTRACT_TOKEN(Strings, 19, '|') AS IP_Origem INTO '" +
    $Destination + "' FROM 'C:\disserta\LPSV2.D1-LH\logs\Security\*.evtx' WHERE EventID
    = 4625 AND Conta_Usuario_Servico NOT IN ('SYSTEM'; 'ANONYMOUS LOGON'; 'LOCAL
    SERVICE'; 'NETWORK SERVICE') ORDER BY Data_Horario DESC"
    $rtnVal = $LogQuery.ExecuteBatch($SQLQuery, $InputFormat, $OutputFormat);
    $OutputFormat = $null;
    $InputFormat = $null;
    $LogQuery = $null;

    if($AutoOpen)
    {
        try
        {
            Start-Process($Destination)
        }
        catch
        {
            Write-Host $_.Exception.Message -ForegroundColor Red
            Write-Host $_.Exception.GetType().FullName -ForegroundColor Red
            Write-Host "NOTE: No output file will be created if the query returned
            zero records!" -ForegroundColor Gray
        }
    }
}

```

F

Query 03 – Encontrar os Eventos 4648 – *Logon* com Credenciais Explícitas

```
/* Encontrar os eventos 4648 - logon com credenciais explícitas */  
  
SELECT TimeGenerated AS Data_Horário,  
       EXTRACT_TOKEN(strings, 2, '|') AS Dominio,  
       EXTRACT_TOKEN(Strings, 5, '|') AS Conta_Usuario_Servico,  
       EXTRACT_TOKEN(Strings, 12, '|') AS IP_Origem  
  
FROM '[LOGFILEPATH]'  
  
WHERE EventID = 4648
```

G

Script PS1 – Encontrar os Eventos 4648

```
#####
# Generated by Log Parser Studio #
#####

#Name: Evento4648
#Log Type: EVTLOG
#Generated: 02/10/2017 15:17:16

Param(
    [parameter(Mandatory=$false)]
    [Bool] $AutoOpen,
    [parameter(Mandatory=$false)]
    [String] $OutFile,
    [parameter(Mandatory=$false)]
    [Bool] $IgnoreInParams,
    [parameter(Mandatory=$false)]
    [Bool] $IgnoreOutParams)

$Error.Clear()
$DefaultFolder=[Environment]::GetFolderPath("MyDocuments")
$Destination = "Evento4648.csv"
$Destination = $DefaultFolder + "\" + $Destination

if($OutFile -ne [String]::Empty)
{
    $OutFileType = [System.IO.Path]::GetExtension($OutFile.ToUpper())
    $OriginalFileType = [System.IO.Path]::GetExtension($Destination.ToUpper())
    if($OutFileType -ne $OriginalFileType)
    {
        Write-Host "You have chosen" $OutFileType "as the output, but this
script was originally generated as" $OriginalFileType -ForegroundColor Red
        Write-Host "Either change -OutFile to" $OriginalFileType "or generate
the script again with the output as" $OutFileType -ForegroundColor Red
        Write-Host "You can also modify the OutputFormat variable in this
script to match the correct Log Parser 2.2 COM output format." -ForegroundColor Red
        [System.Environment]::NewLine
        return
    }
    else
    {
        if($true -ne $OutFile.Contains("\"))
        {
            $Destination = $DefaultFolder + "\" + $OutFile
        }
        else
        {
            $Destination = $OutFile
        }
    }
}

$LogQuery = New-Object -ComObject "MSUtil.LogQuery"
$InputFormat = New-Object -ComObject "MSUtil.LogQuery.EventLogInputFormat"
$OutputFormat = New-Object -ComObject "MSUtil.LogQuery.CSVOutputFormat"

if($IgnoreInParams-eq $false){
    $InputFormat.fullText=1
    $InputFormat.resolveSIDs=0
    $InputFormat.formatMsg=1
    $InputFormat.msgErrorMode="MSG"
    $InputFormat.fullEventCode=0
    $InputFormat.direction="FW"
    $InputFormat.stringsSep="|"
    $InputFormat.binaryFormat="PRINT"
}
```

```

        $InputFormat.ignoreMessageErrors=1
    }
    if($IgnoreOutParams -eq $false){
        $OutputFormat.Headers="AUTO"
        $OutputFormat.oDQuotes="AUTO"
        $OutputFormat.tabs=0
        $OutputFormat.oTsFormat="yyyy-MM-dd hh:mm:ss"
        $OutputFormat.oCodepage=0
        $OutputFormat.fileMode=1
    }
    Write-Progress -Activity "Executing query, please wait..." -Status " "

    $SQLQuery = "SELECT TimeGenerated AS Data_Horário, EXTRACT_TOKEN(strings, 2, '|')
    AS Dominio, EXTRACT_TOKEN(Strings, 5, '|') AS Conta_Usuario_Servico,
    EXTRACT_TOKEN(Strings, 12, '|') AS IP_Origem INTO '"' + $Destination + '"' FROM
    'C:\disserta\LPSV2.D1-LH\logs\Security\*.evtx' WHERE EventID = 4648"
    $rtnVal = $LogQuery.ExecuteBatch($SQLQuery, $InputFormat, $OutputFormat);
    $OutputFormat = $null;
    $InputFormat = $null;
    $LogQuery = $null;

    if($AutoOpen)
    {
        try
        {
            Start-Process($Destination)
        }
        catch
        {
            Write-Host $_.Exception.Message -ForegroundColor Red
            Write-Host $_.Exception.GetType().FullName -ForegroundColor Red
            Write-Host "NOTE: No output file will be created if the query returned
            zero records!" -ForegroundColor Gray
        }
    }
}

```

H

Query 04 – Encontrar os Eventos 4660 E 4663 – Exclusão de Arquivos

```
/* Consulta dos Eventos 4660 e 4663 - Exclusao de objetos monitorados */  
  
SELECT TimeGenerated AS Data_Horario, EventID AS ID_Evento,  
       EXTRACT_TOKEN(Strings, 1, '|') AS Conta_Usuario_Servico,  
       EXTRACT_TOKEN(Strings, 2, '|') AS Dominio,  
       EXTRACT_TOKEN(Strings, 4, '|') AS Codigo_Bloqueio,  
       EXTRACT_TOKEN(Strings, 5, '|') AS Tipo_Objeto,  
       EXTRACT_TOKEN(Strings, 6, '|') AS Objeto_Acessado,  
       EXTRACT_TOKEN(Strings, 7, '|') AS SubCodigo  
  
FROM '[LOGFILEPATH]'  
WHERE EventID = 4660 OR EventID = 4663
```

I

Script PS1– Encontrar os Eventos 4660 E 4663

```
#####
# Generated by Log Parser Studio #
#####

#Name: Eventos4660_4663
#Log Type: EVTLOG
#Generated: 02/10/2017 15:27:34

Param(
    [parameter(Mandatory=$false)]
    [Bool] $AutoOpen,
    [parameter(Mandatory=$false)]
    [String] $OutFile,
    [parameter(Mandatory=$false)]
    [Bool] $IgnoreInParams,
    [parameter(Mandatory=$false)]
    [Bool] $IgnoreOutParams)

$Error.Clear()
$DefaultFolder=[Environment]::GetFolderPath("MyDocuments")
$Destination = "Eventos4660_4663.csv"
$Destination = $DefaultFolder + "\" + $Destination

if($OutFile -ne [String]::Empty)
{
    $OutFileType = [System.IO.Path]::GetExtension($OutFile.ToUpper())
    $OriginalFileType = [System.IO.Path]::GetExtension($Destination.ToUpper())
    if($OutFileType -ne $OriginalFileType)
    {
        Write-Host "You have chosen" $OutFileType "as the output, but this
script was originally generated as" $OriginalFileType -ForegroundColor Red
        Write-Host "Either change -OutFile to" $OriginalFileType "or generate
the script again with the output as" $OutFileType -ForegroundColor Red
        Write-Host "You can also modify the OutputFormat variable in this
script to match the correct Log Parser 2.2 COM output format." -ForegroundColor Red
        [System.Environment]::NewLine
        return
    }
    else
    {
        if($true -ne $OutFile.Contains("\"))
        {
            $Destination = $DefaultFolder + "\" + $OutFile
        }
        else
        {
            $Destination = $OutFile
        }
    }
}

$LogQuery = New-Object -ComObject "MSUtil.LogQuery"
$InputFormat = New-Object -ComObject "MSUtil.LogQuery.EventLogInputFormat"
$OutputFormat = New-Object -ComObject "MSUtil.LogQuery.CSVOutputFormat"

if($IgnoreInParams-eq $false){
    $InputFormat.fullText=1
    $InputFormat.resolveSIDs=0
    $InputFormat.formatMsg=1
    $InputFormat.msgErrorMode="MSG"
    $InputFormat.fullEventCode=0
    $InputFormat.direction="Fw"
    $InputFormat.stringsSep="|"
    $InputFormat.binaryFormat="PRINT"
```

```

    $InputFormat.ignoreMessageErrors=1
}
if($IgnoreOutParams -eq $false){
    $OutputFormat.Headers="AUTO"
    $OutputFormat.oDQuotes="AUTO"
    $OutputFormat.tabs=0
    $OutputFormat.oTsFormat="yyyy-MM-dd hh:mm:ss"
    $OutputFormat.oCodepage=0
    $OutputFormat.fileMode=1
}
Write-Progress -Activity "Executing query, please wait..." -Status " "

$SQLQuery = "SELECT TimeGenerated AS Data_Horario, EventID AS ID_Evento,
EXTRACT_TOKEN(Strings, 1, '|') AS Conta_Usuario_Servico, EXTRACT_TOKEN(Strings, 2,
'|') AS Dominio, EXTRACT_TOKEN(Strings, 4, '|') AS Codigo_Bloqueio,
EXTRACT_TOKEN(Strings, 5, '|') AS Tipo_Objeto, EXTRACT_TOKEN(Strings, 6, '|') AS
Objeto_Acessado, EXTRACT_TOKEN(Strings, 7, '|') AS SubCodigo INTO '" + $Destination
+ "' FROM 'C:\disserta\LPSV2.D1-LH\logs\Security\*.evtx' WHERE EventID = 4660 OR
EventID = 4663"
$rtVal = $LogQuery.ExecuteBatch($SQLQuery, $InputFormat, $OutputFormat);
$OutputFormat = $null;
$InputFormat = $null;
$LogQuery = $null;

if($AutoOpen)
{
    try
    {
        Start-Process($Destination)
    }
    catch
    {
        Write-Host $_.Exception.Message -ForegroundColor Red
        Write-Host $_.Exception.GetType().FullName -ForegroundColor Red
        Write-Host "NOTE: No output file will be created if the query returned
zero records!" -ForegroundColor Gray
    }
}

```

J

Query 05 – Encontrar os Eventos 4771 – Contas Bloqueadas

```
/* Consulta dos Eventos 4771 - Contas bloqueadas */  
  
SELECT TimeGenerated AS Data_Horario,  
       EXTRACT_TOKEN(Strings, 0, '|') AS Conta_Usuario_Servico,  
       EXTRACT_TOKEN(Strings, 4, '|') AS Codigo_Bloqueio,  
       EXTRACT_TOKEN(Strings, 6, '|') AS IP_Estacao  
  
INTO 'Evento4771.csv'  
FROM '[LOGFILEPATH]'  
WHERE EventID = 4771
```

K

Script PS1 – Encontrar os Eventos 4771

```
#####
# Generated by Log Parser Studio #
#####

#Name: Evento4771
#Log Type: EVTLOG
#Generated: 01/10/2017 14:53:42

Param(
    [parameter(Mandatory=$false)]
    [Bool] $AutoOpen,
    [parameter(Mandatory=$false)]
    [String] $OutFile,
    [parameter(Mandatory=$false)]
    [Bool] $IgnoreInParams,
    [parameter(Mandatory=$false)]
    [Bool] $IgnoreOutParams)

$Error.Clear()
$DefaultFolder=[Environment]::GetFolderPath("MyDocuments")
$Destination = "Evento4771.csv"
$Destination = $DefaultFolder + "\" + $Destination

if($OutFile -ne [String]::Empty)
{
    $OutFileType = [System.IO.Path]::GetExtension($OutFile.ToUpper())
    $OriginalFileType = [System.IO.Path]::GetExtension($Destination.ToUpper())
    if($OutFileType -ne $OriginalFileType)
    {
        Write-Host "You have chosen" $OutFileType "as the output, but this
script was originally generated as" $OriginalFileType -ForegroundColor Red
        Write-Host "Either change -OutFile to" $OriginalFileType "or generate
the script again with the output as" $OutFileType -ForegroundColor Red
        Write-Host "You can also modify the OutputFormat variable in this
script to match the correct Log Parser 2.2 COM output format." -ForegroundColor Red
        [System.Environment]::NewLine
        return
    }
    else
    {
        if($true -ne $OutFile.Contains("\"))
        {
            $Destination = $DefaultFolder + "\" + $OutFile
        }
        else
        {
            $Destination = $OutFile
        }
    }
}

$LogQuery = New-Object -ComObject "MSUtil.LogQuery"
$InputFormat = New-Object -ComObject "MSUtil.LogQuery.EventLogInputFormat"
$OutputFormat = New-Object -ComObject "MSUtil.LogQuery.CSVOutputFormat"

if($IgnoreInParams-eq $false){
    $InputFormat.fullText=1
    $InputFormat.resolveSIDs=0
    $InputFormat.formatMsg=1
    $InputFormat.msgErrorMode="MSG"
    $InputFormat.fullEventCode=0
    $InputFormat.direction="Fw"
    $InputFormat.stringsSep="|"
    $InputFormat.binaryFormat="PRINT"
}
```

```

        $InputFormat.ignoreMessageErrors=1
    }
    if($IgnoreOutParams -eq $false){
        $OutputFormat.Headers="AUTO"
        $OutputFormat.oDQuotes="AUTO"
        $OutputFormat.tabs=0
        $OutputFormat.oTsFormat="yyyy-MM-dd hh:mm:ss"
        $OutputFormat.oCodepage=0
        $OutputFormat.fileMode=1
    }
    Write-Progress -Activity "Executing query, please wait..." -Status " "

    $SQLQuery = "SELECT TimeGenerated AS Data_Horario, EXTRACT_TOKEN(Strings, 0, '|')
    AS Conta_Usuario_Servico, EXTRACT_TOKEN(Strings, 4, '|') AS Codigo_Bloqueio,
    EXTRACT_TOKEN(Strings, 6, '|') AS IP_Estacao INTO '" + $Destination + "' FROM
    'C:\disserta\LPSV2.D1-LH\logs\Security\*.evtx' WHERE EventID = 4771"
    $rtnVal = $LogQuery.ExecuteBatch($SQLQuery, $InputFormat, $OutputFormat);
    $OutputFormat = $null;
    $InputFormat = $null;
    $LogQuery = $null;

    if($AutoOpen)
    {
        try
        {
            Start-Process($Destination)
        }
        catch
        {
            Write-Host $_.Exception.Message -ForegroundColor Red
            Write-Host $_.Exception.GetType().FullName -ForegroundColor Red
            Write-Host "NOTE: No output file will be created if the query returned
            zero records!" -ForegroundColor Gray
        }
    }
}

```